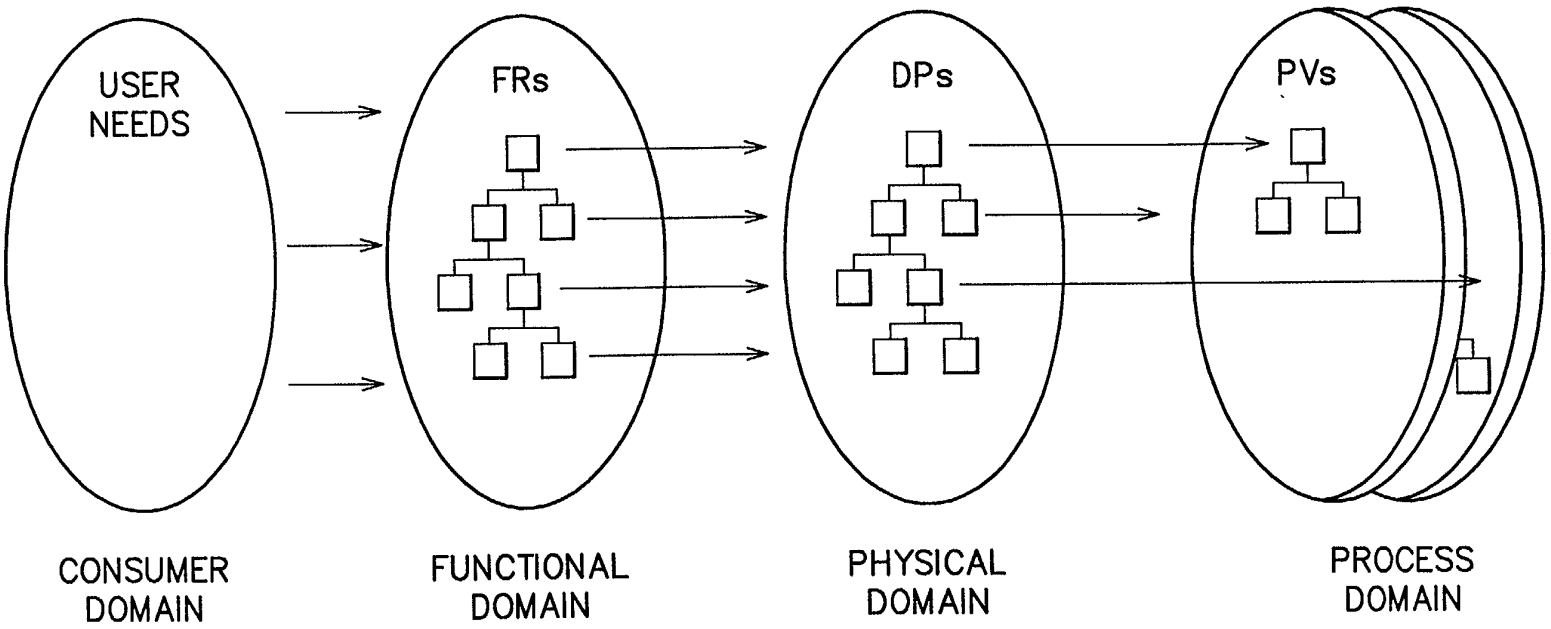


FIG. 1

UNCOUPLED – Desired

	DP1	DP2	DP3
FR1	X	0	0
FR2	0	X	0
FR3	0	0	X

DECOUPLED – Acceptable

	DP1	DP2	DP3
FR1	X	0	0
FR2	X	X	0
FR3	X	X	X

COUPLED – Undesired

	DP1	DP2	DP3
FR1	X	X	X
FR2	X	X	X
FR3	X	X	X

FIG. 2

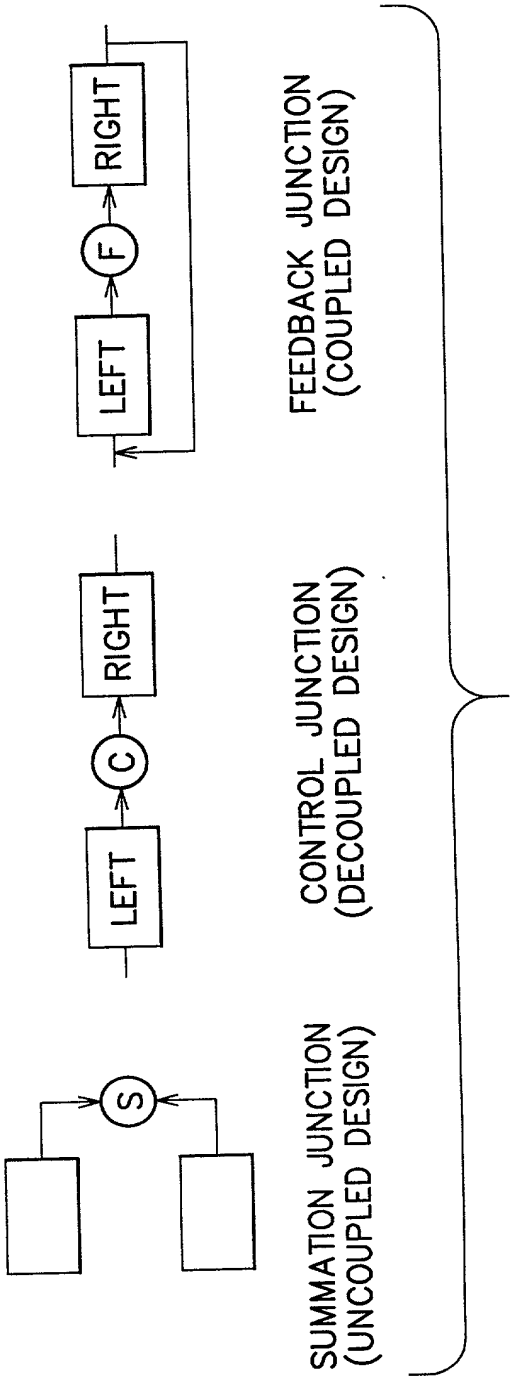

SUMMATION JUNCTION
(UNCOUPLED DESIGN)
S
CONTROL JUNCTION
(DECOUPLED DESIGN)
LEFT
C
RIGHT
FEEDBACK JUNCTION
(COUPLED DESIGN)
LEFT
F
RIGHT

FIG. 3

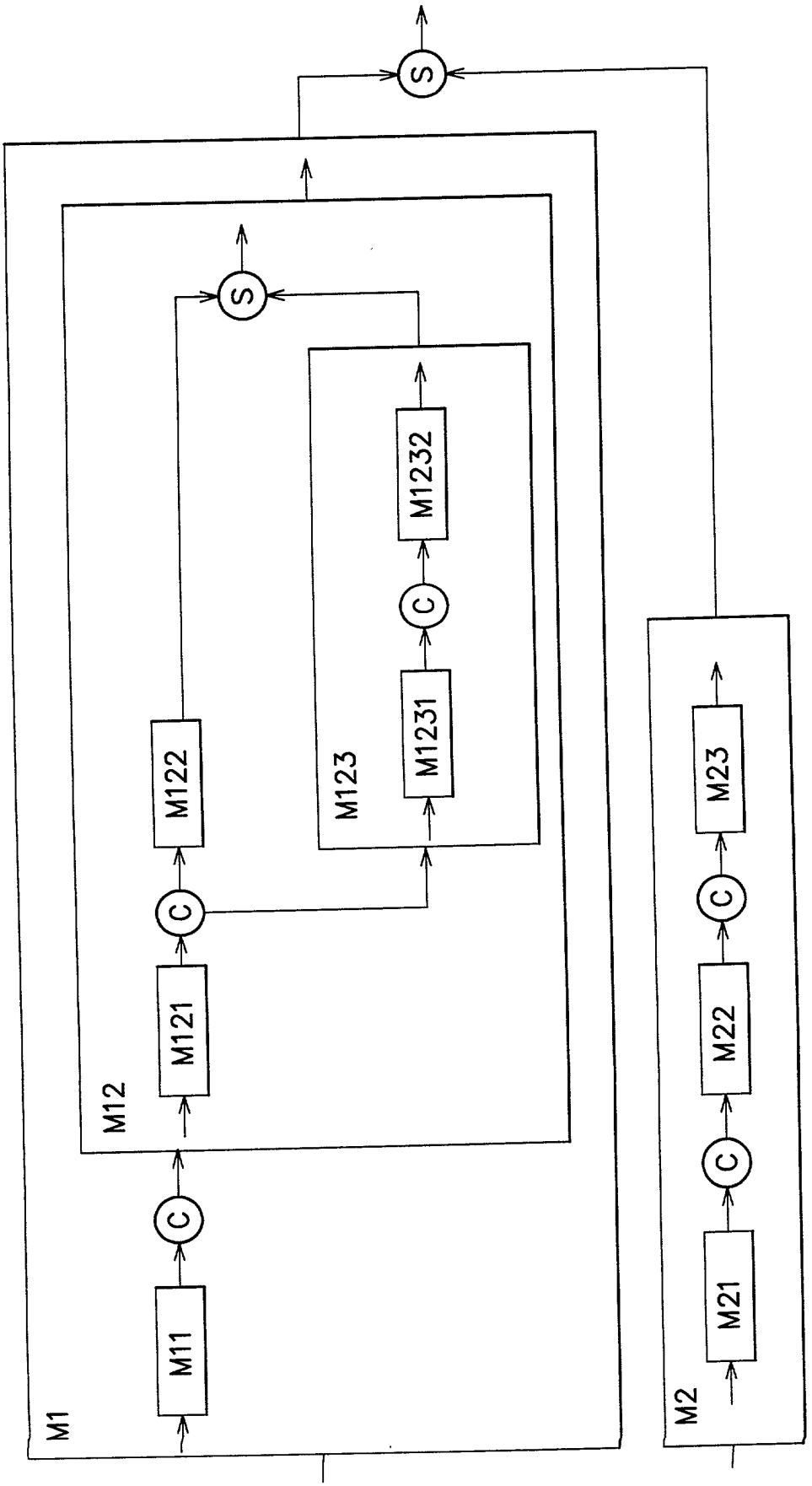


FIG. 4

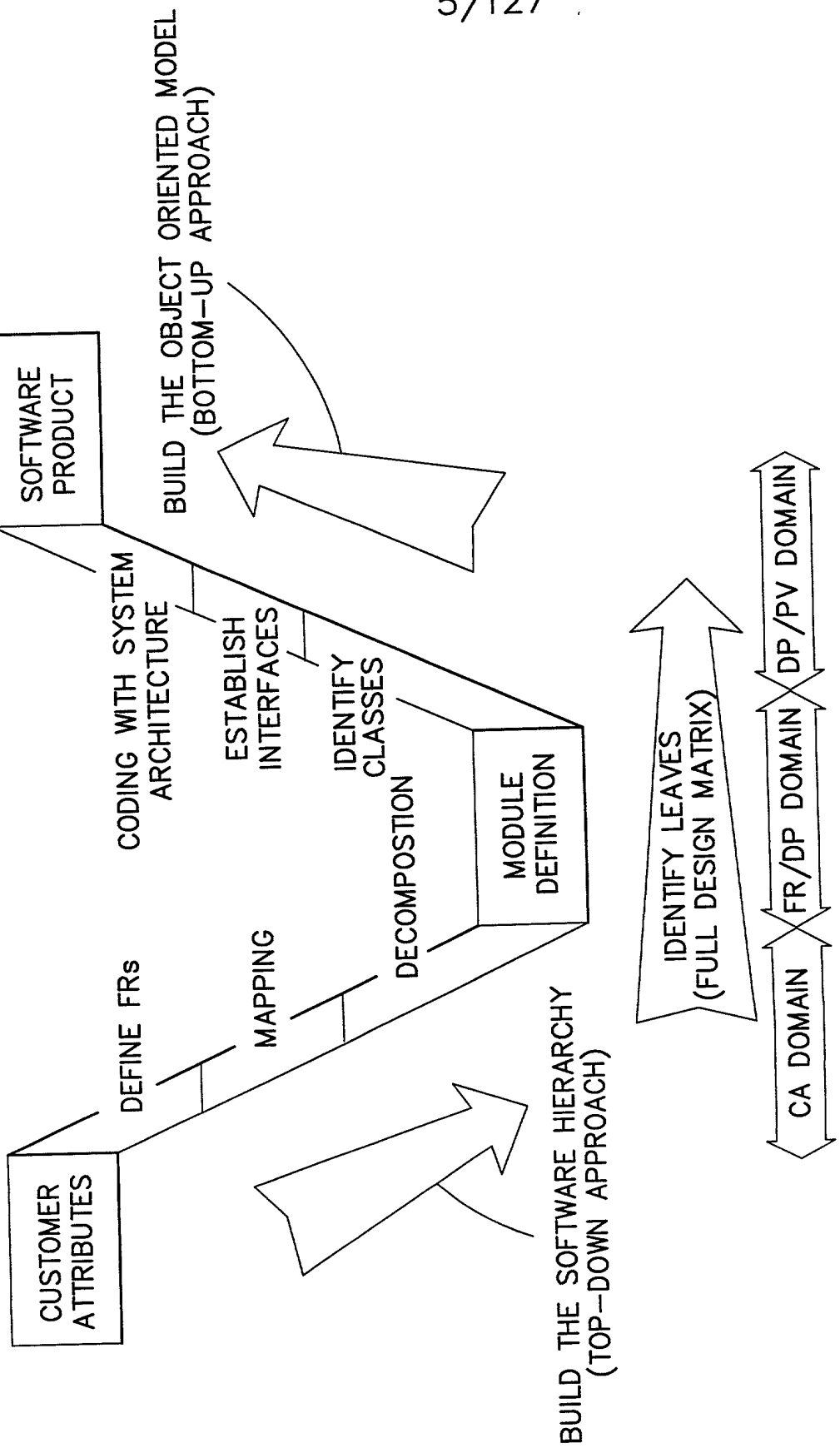

SOFTWARE PRODUCT
BUILD THE OBJECT ORIENTED MODEL (BOTTOM–UP APPROACH)
CODING WITH SYSTEM ARCHITECTURE
ESTABLISH INTERFACES
IDENTIFY CLASSES
MODULE DEFINITION
DECOMPOSTION
MAPPING
DEFINE FRs
CUSTOMER ATTRIBUTES
BUILD THE SOFTWARE HIERARCHY (TOP–DOWN APPROACH)
IDENTIFY LEAVES (FULL DESIGN MATRIX)
CA DOMAIN
FR/DP DOMAIN
DP/PV DOMAIN

FIG. 5

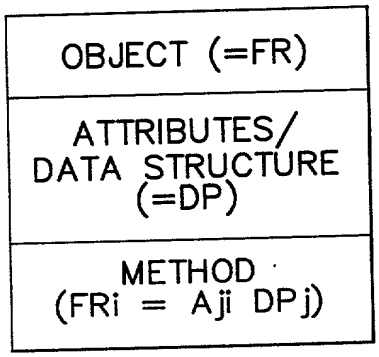


FIG. 6

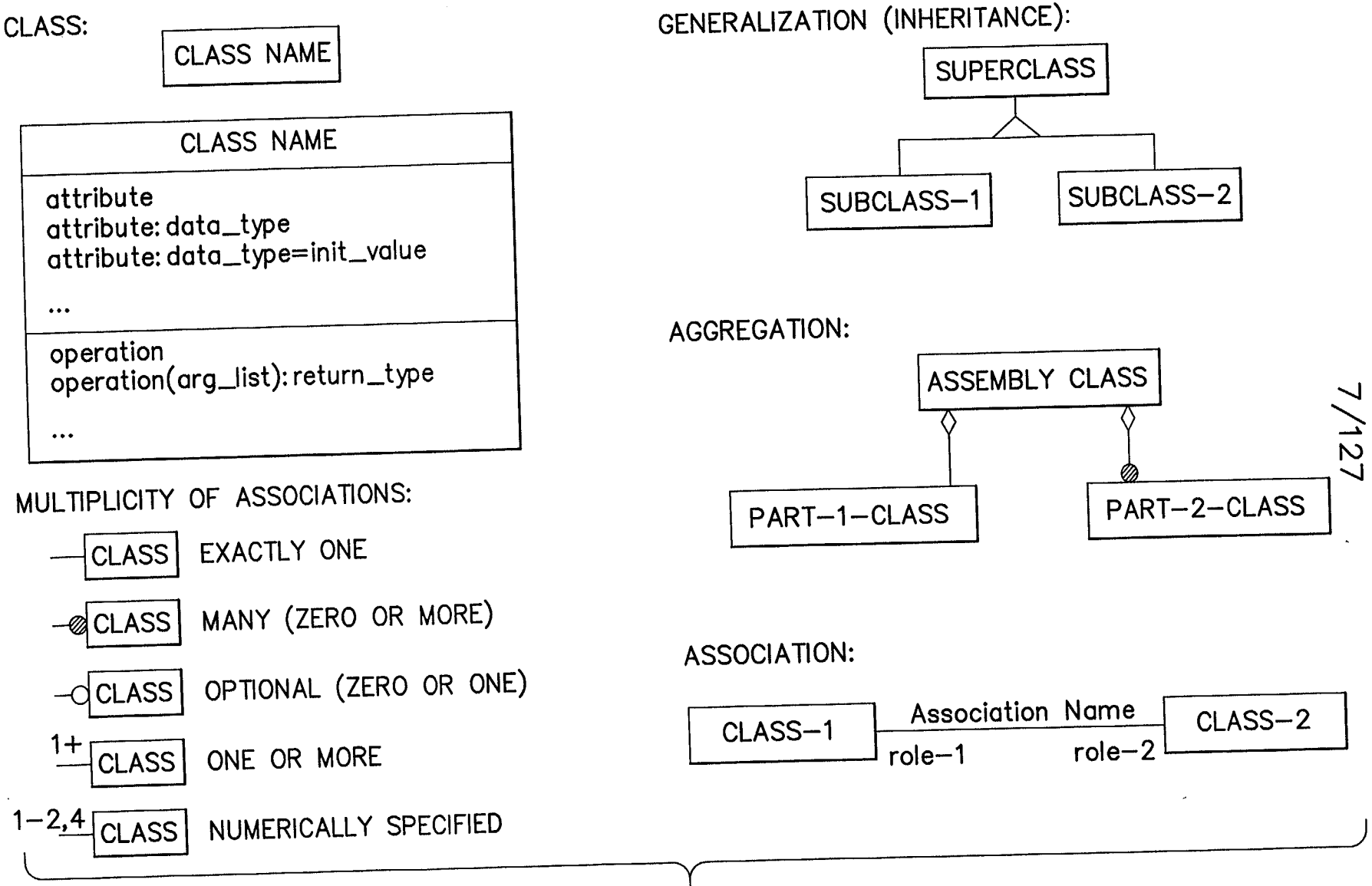

CLASS:
CLASS NAME
CLASS NAME
attribute
attribute: data_type
attribute: data_type=init_value
...
operation
operation(arg_list): return_type
...
MULTIPLICITY OF ASSOCIATIONS:
CLASS EXACTLY ONE
CLASS MANY (ZERO OR MORE)
CLASS OPTIONAL (ZERO OR ONE)
1+ CLASS ONE OR MORE
1-2,4 CLASS NUMERICALLY SPECIFIED
GENERALIZATION (INHERITANCE):
SUPERCLASS
SUBCLASS-1
SUBCLASS-2
AGGREGATION:
ASSEMBLY CLASS
PART-1-CLASS
PART-2-CLASS
ASSOCIATION:
CLASS-1
Association Name
role-1
role-2
CLASS-2

FIG. 7

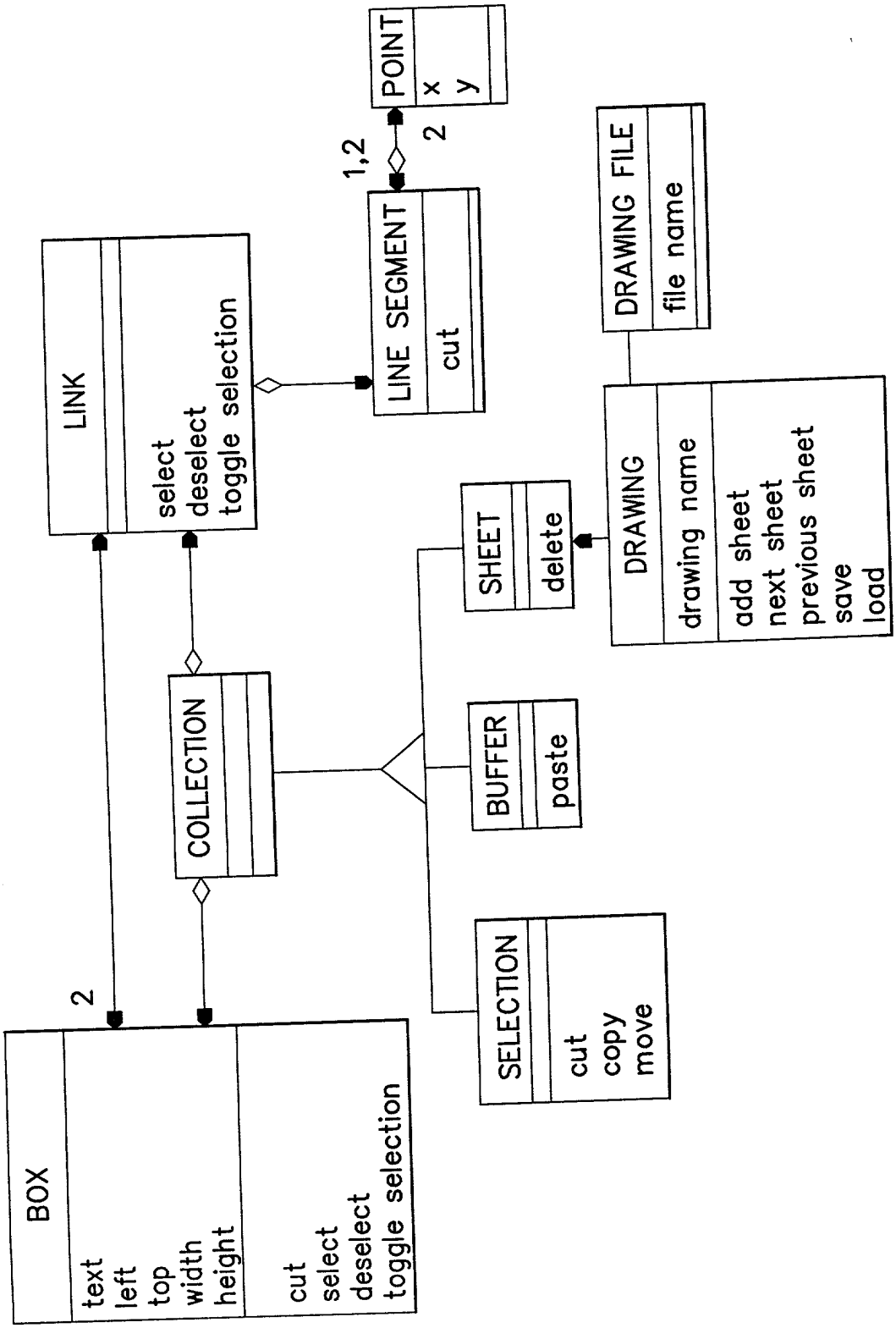

BOX
text
left
top
width
height
cut
select
deselect
toggle selection
2
LINK
select
deselect
toggle selection
COLLECTION
SELECTION
cut
copy
move
BUFFER
paste
SHEET
delete
DRAWING
drawing name
add sheet
next sheet
previous sheet
save
load
DRAWING FILE
file name
LINE SEGMENT
cut
1,2
2
POINT
x
y

FIG. 8

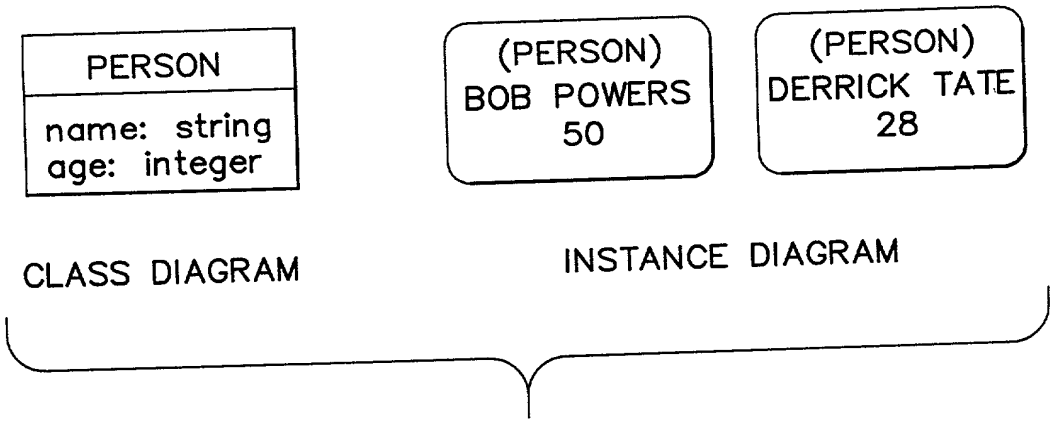

PERSON
name: string
age: integer
(PERSON)
BOB POWERS
50
(PERSON)
DERRICK TATE
28
CLASS DIAGRAM
INSTANCE DIAGRAM

FIG. 9

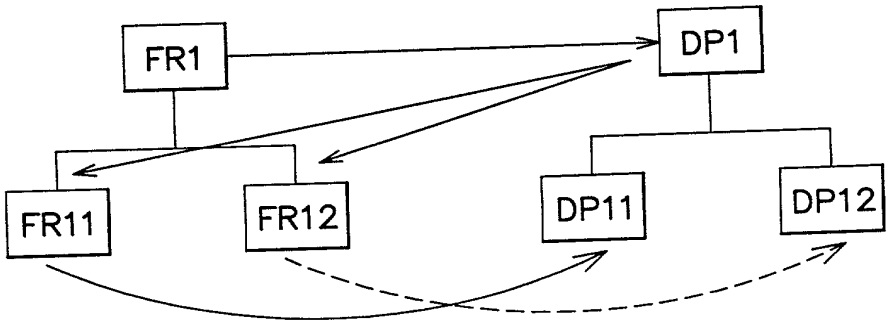

FR1
DP1
FR11
FR12
DP11
DP12

FIG. 10

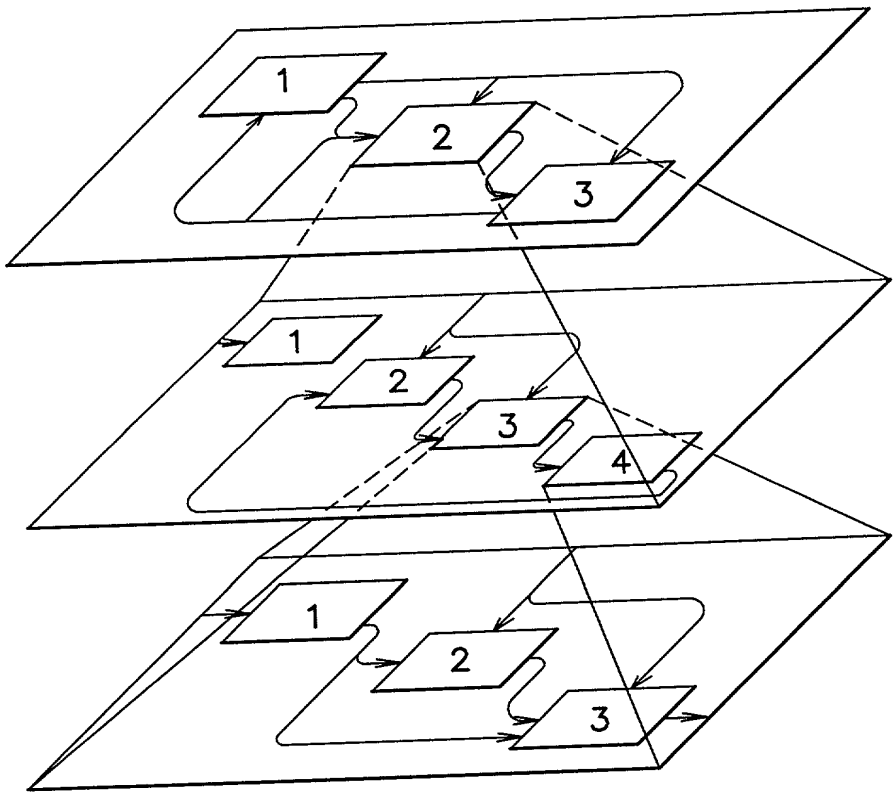


FIG. 11

$$\begin{Bmatrix} \text{FR1: Define Component} \\ \text{FR2: Set Something} \\ \text{FR3: Do Something} \end{Bmatrix} = \begin{bmatrix} X & O & O \\ X & X & O \\ X & X & X \end{bmatrix} \begin{Bmatrix} \text{DP1: Attributes} \\ \text{DP2: Action A} \\ \text{DP3: Action B} \end{Bmatrix}$$

Module Definition

The number of total attributes are 4 for this class.

These two attributes are used by all methods.

	DP1: Attributes	DP2: Action A	DP3: Action B		
FR1: Define component	uPortPinsUp, uPinsUpSensor				
FR2: Set something	uStatus	SetState()		←	Module for FR2
FR3: Do something	uTime	X	ProcessLoop()	←	Module for FR3

This attribute is only used by SetState() method.

ProcessLoop() method calls SetState() method.

FIG. 12

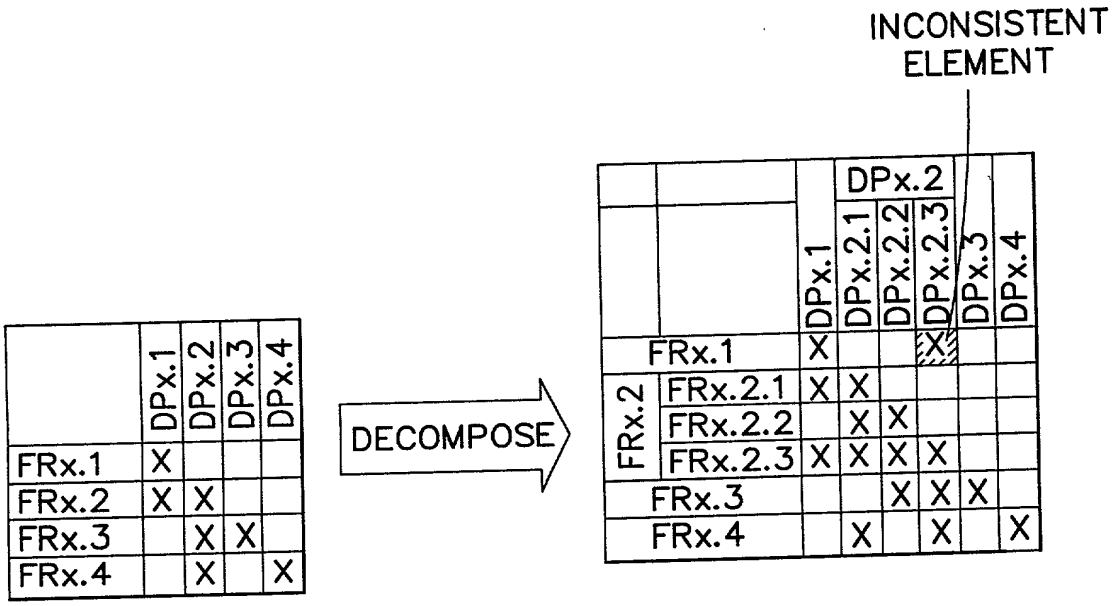

INCONSISTENT ELEMENT
DECOMPOSE
DPx.1
DPx.2
DPx.3
DPx.4
FRx.1
FRx.2
FRx.3
FRx.4
DPx.2.1
DPx.2.2
DPx.2.3
FRx.2.1
FRx.2.2
FRx.2.3

FIG. 13

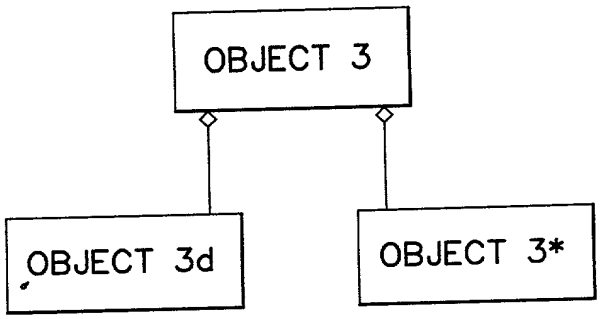

OBJECT 3
OBJECT 3d
OBJECT 3*

FIG. 14

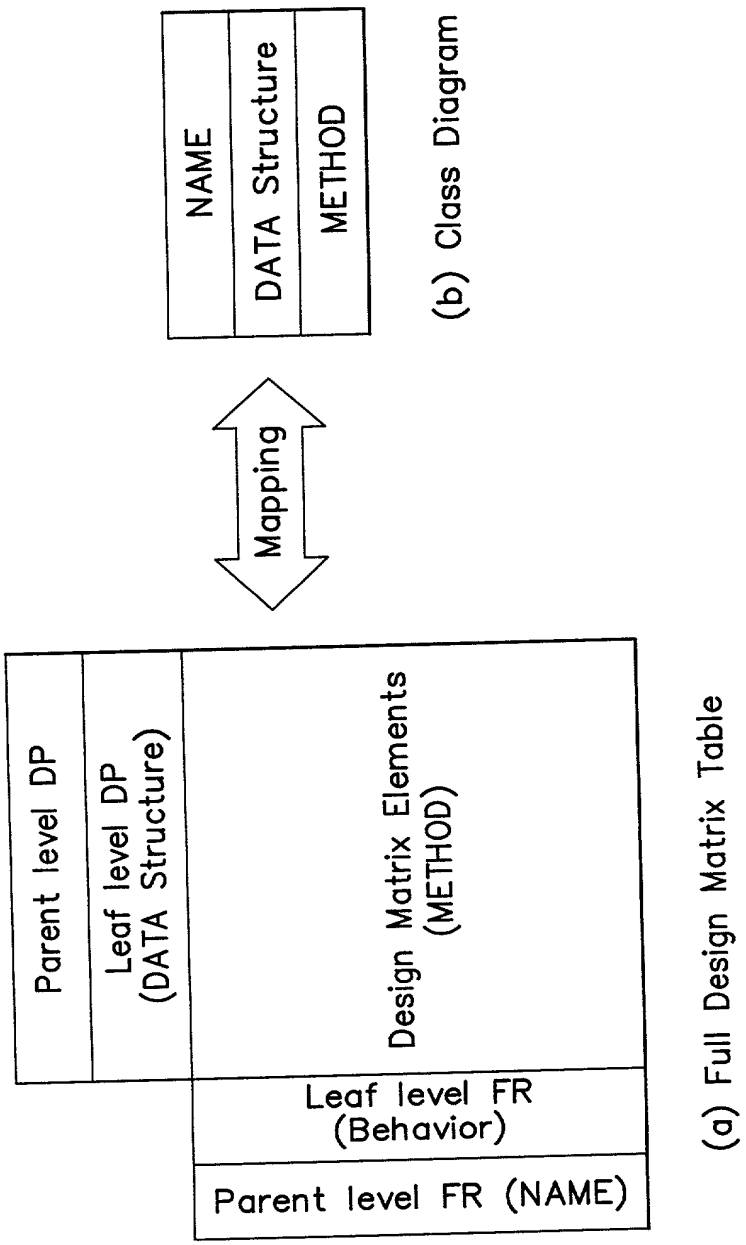

NAME
DATA Structure
METHOD
(b) Class Diagram
Mapping
Parent level DP
Leaf level DP
(DATA Structure)
Design Matrix Elements
(METHOD)
Leaf level FR
(Behavior)
Parent level FR (NAME)
(a) Full Design Matrix Table

FIG. 15

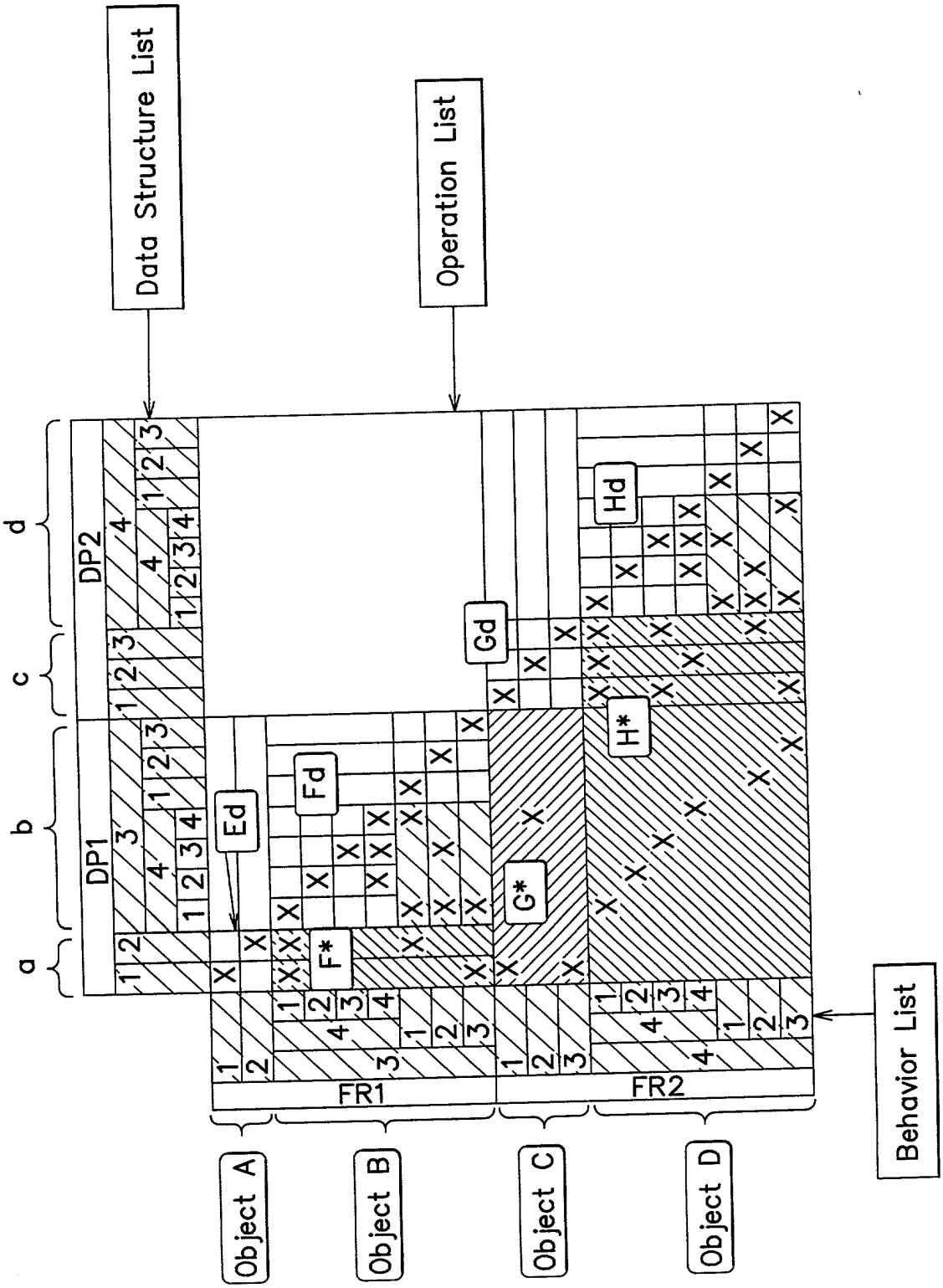

Data Structure List
Operation List
Behavior List
DP1
DP2
a
b
c
d
Ed
Fd
F*
Gd
G*
Hd
H*
FR1
FR2
Object A
Object B
Object C
Object D

FIG. 16

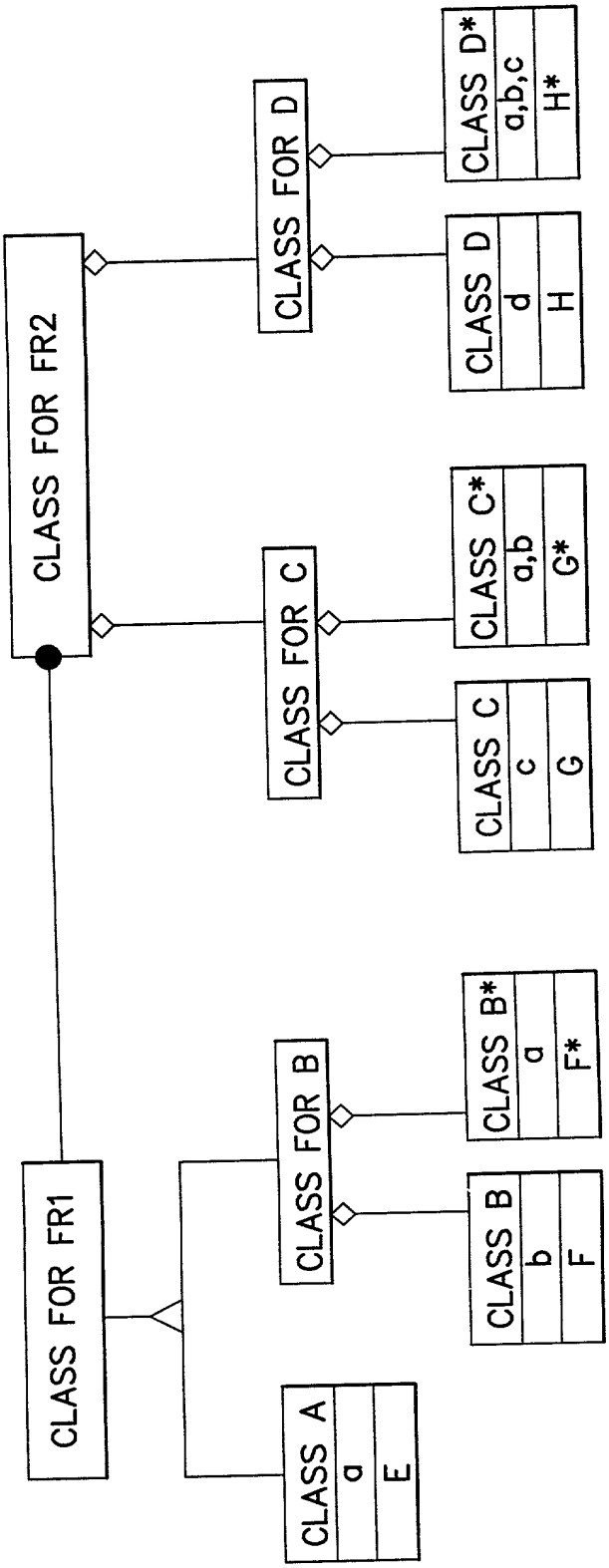

CLASS FOR FR1
CLASS FOR FR2
CLASS FOR B
CLASS FOR C
CLASS FOR D
CLASS A
a
E
CLASS B
b
F
CLASS B*
a
F*
CLASS C
c
G
CLASS C*
a,b
G*
CLASS D
d
H
CLASS D*
a,b,c
H*

FIG. 17

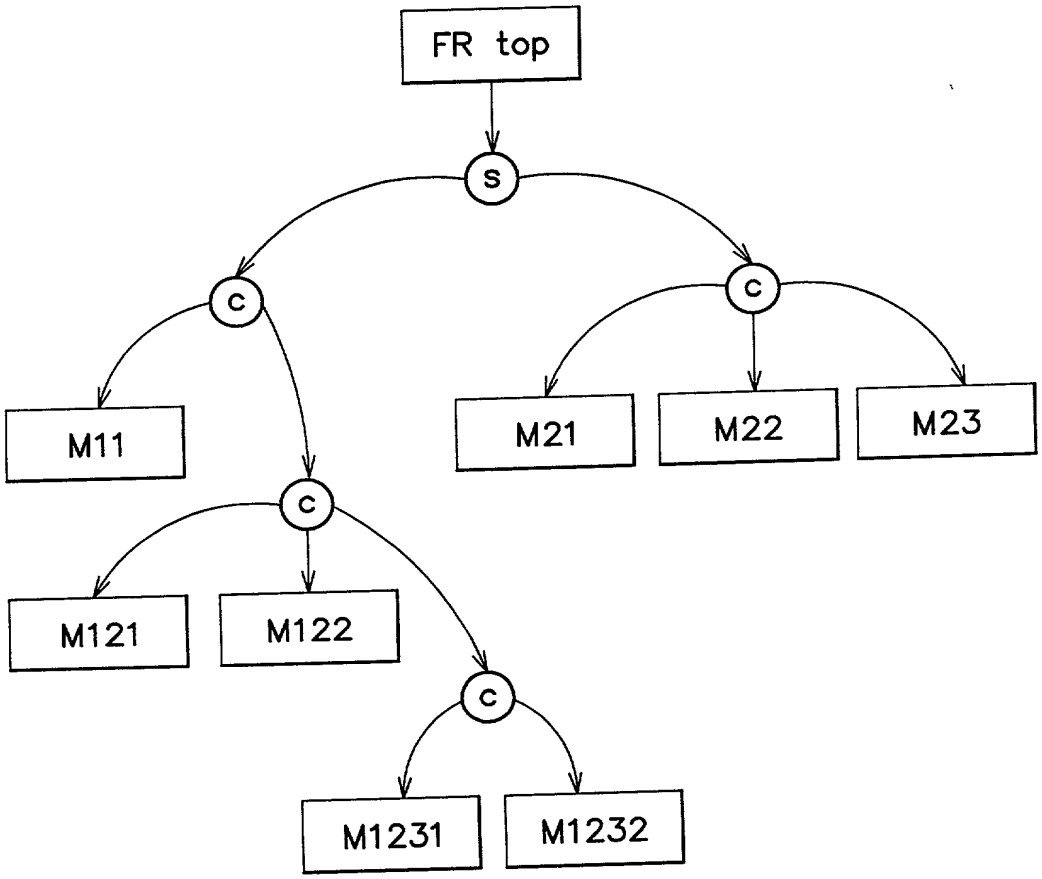


FIG. 18

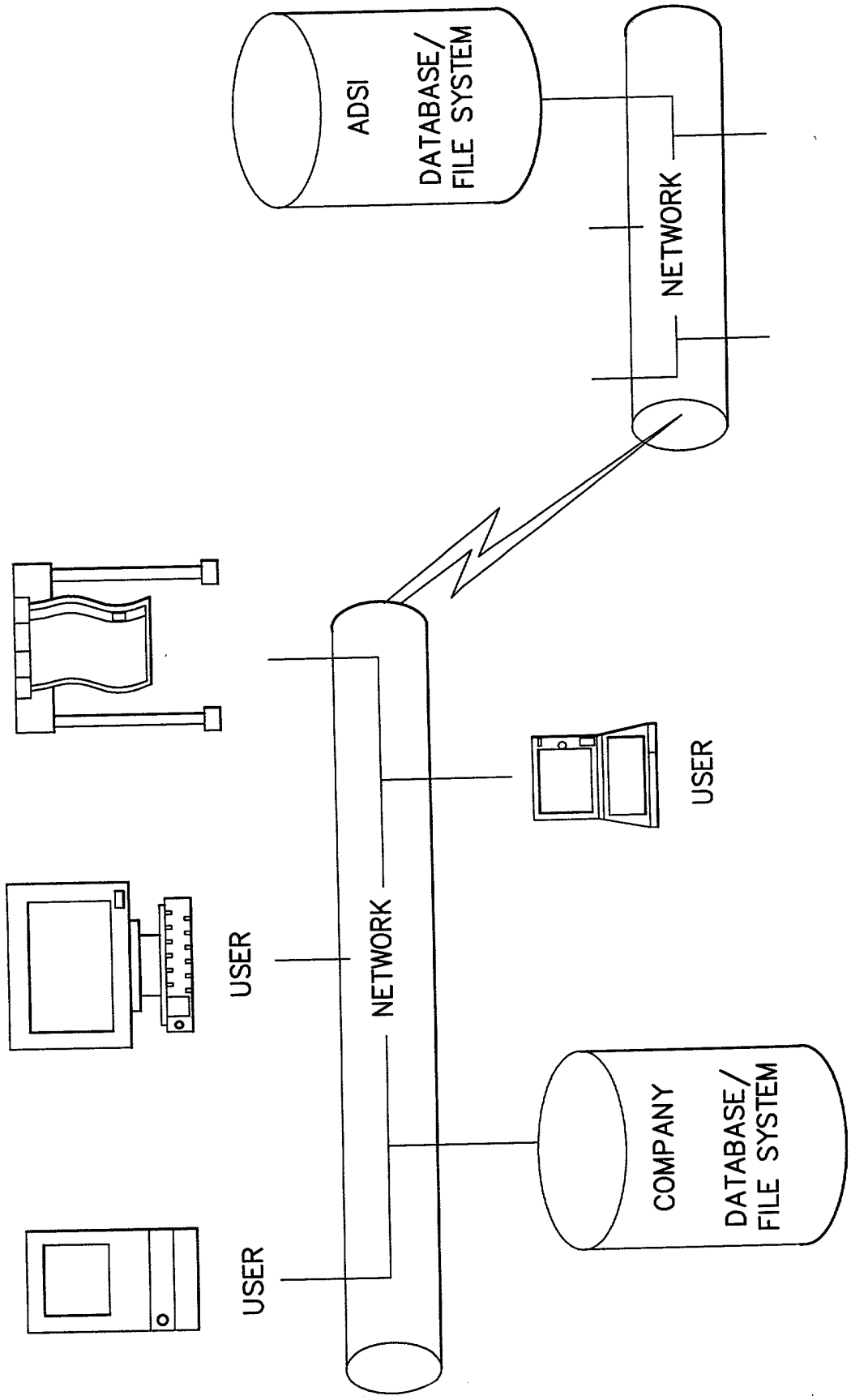

ADSI
DATABASE/
FILE SYSTEM
NETWORK
USER
USER
NETWORK
USER
COMPANY
DATABASE/
FILE SYSTEM

FIG. 19

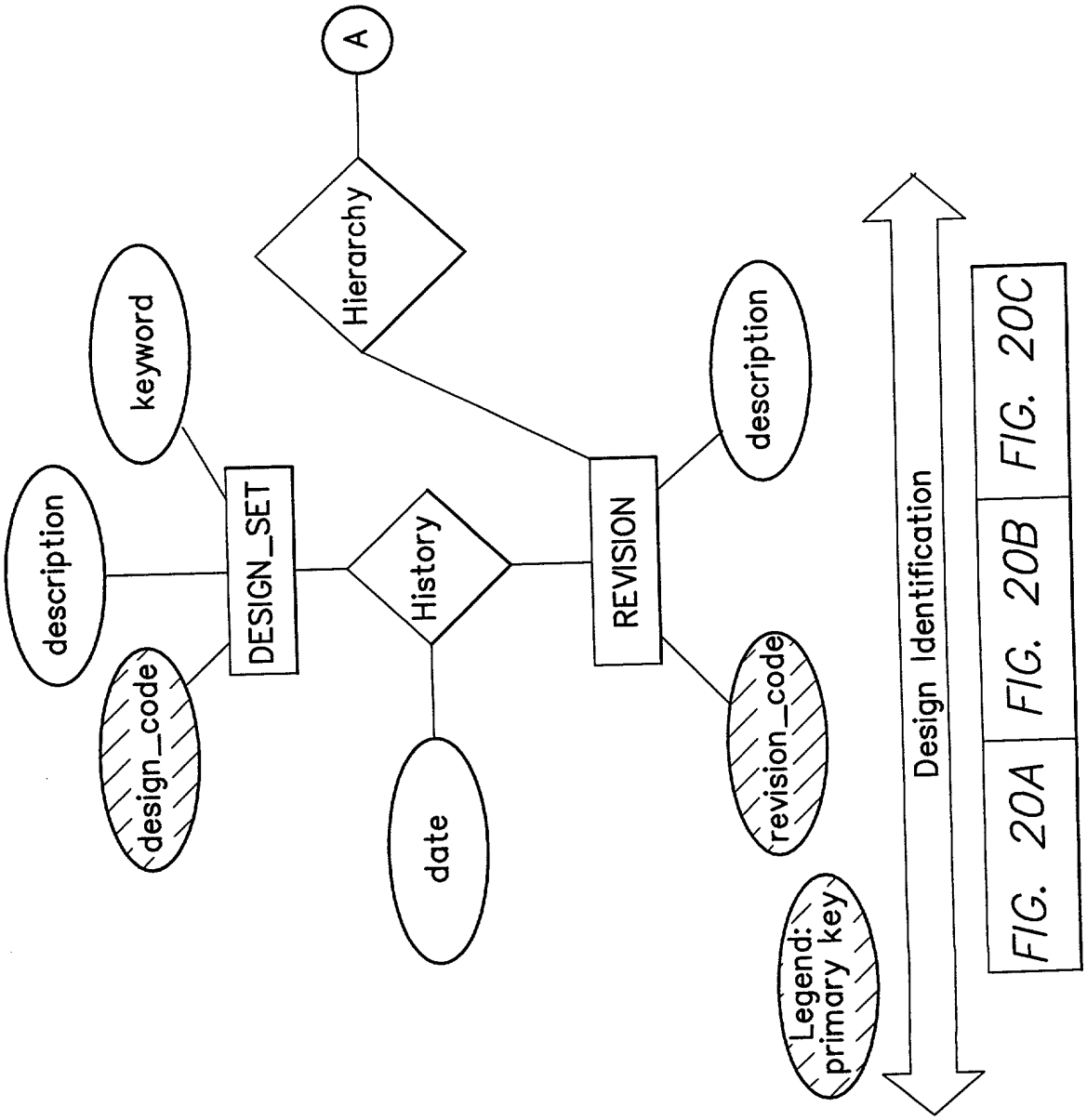

A
Hierarchy
keyword
description
DESIGN_SET
History
REVISION
description
design_code
date
revision_code
Legend:
primary key
Design Identification
FIG. 20A
FIG. 20B
FIG. 20C

FIG. 20A

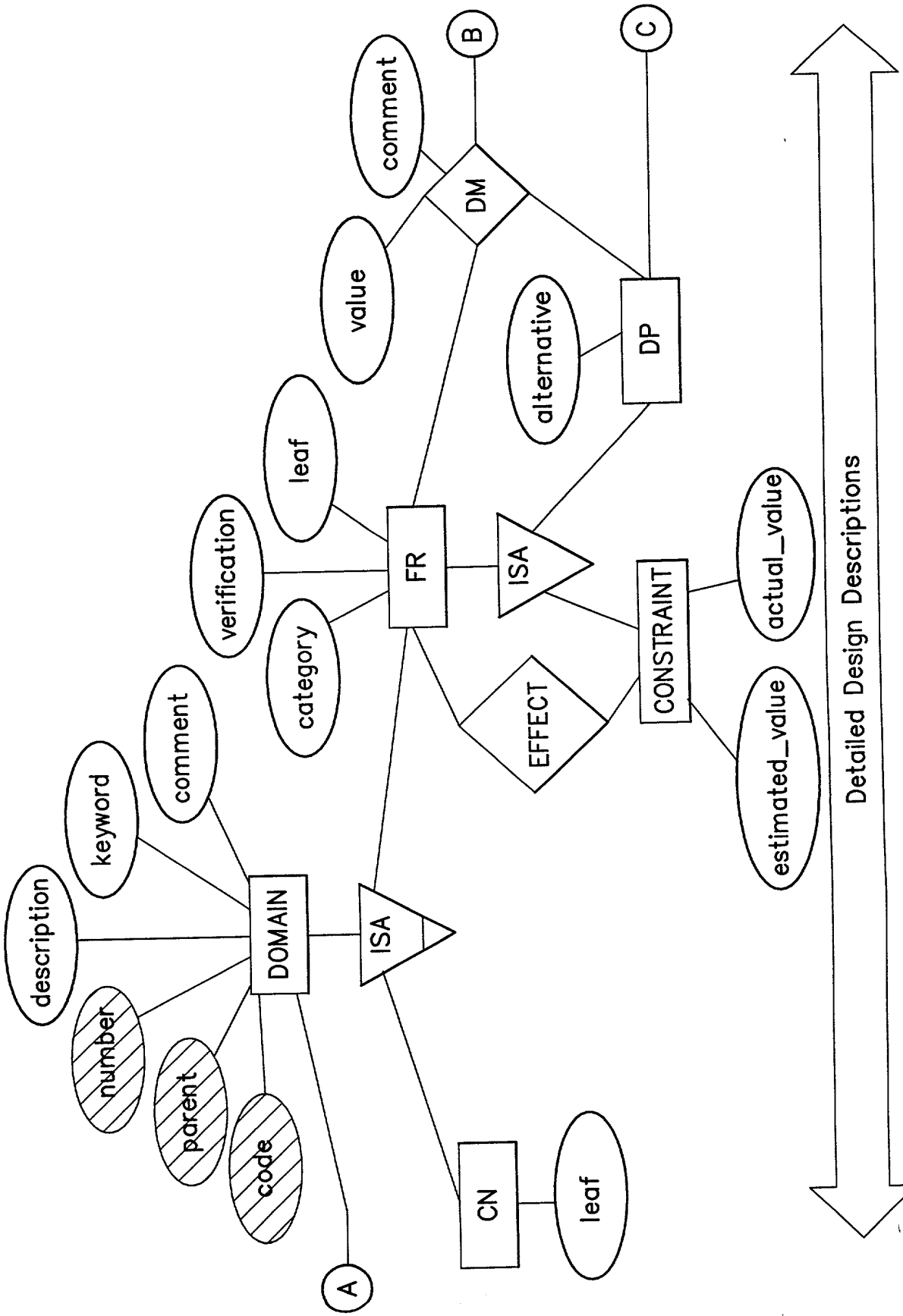

description
keyword
comment
number
parent
code
DOMAIN
ISA
A
CN
leaf
verification
leaf
category
FR
value
comment
DM
B
ISA
EFFECT
alternative
DP
C
CONSTRAINT
actual_value
estimated_value
Detailed Design Descriptions

FIG. 20B

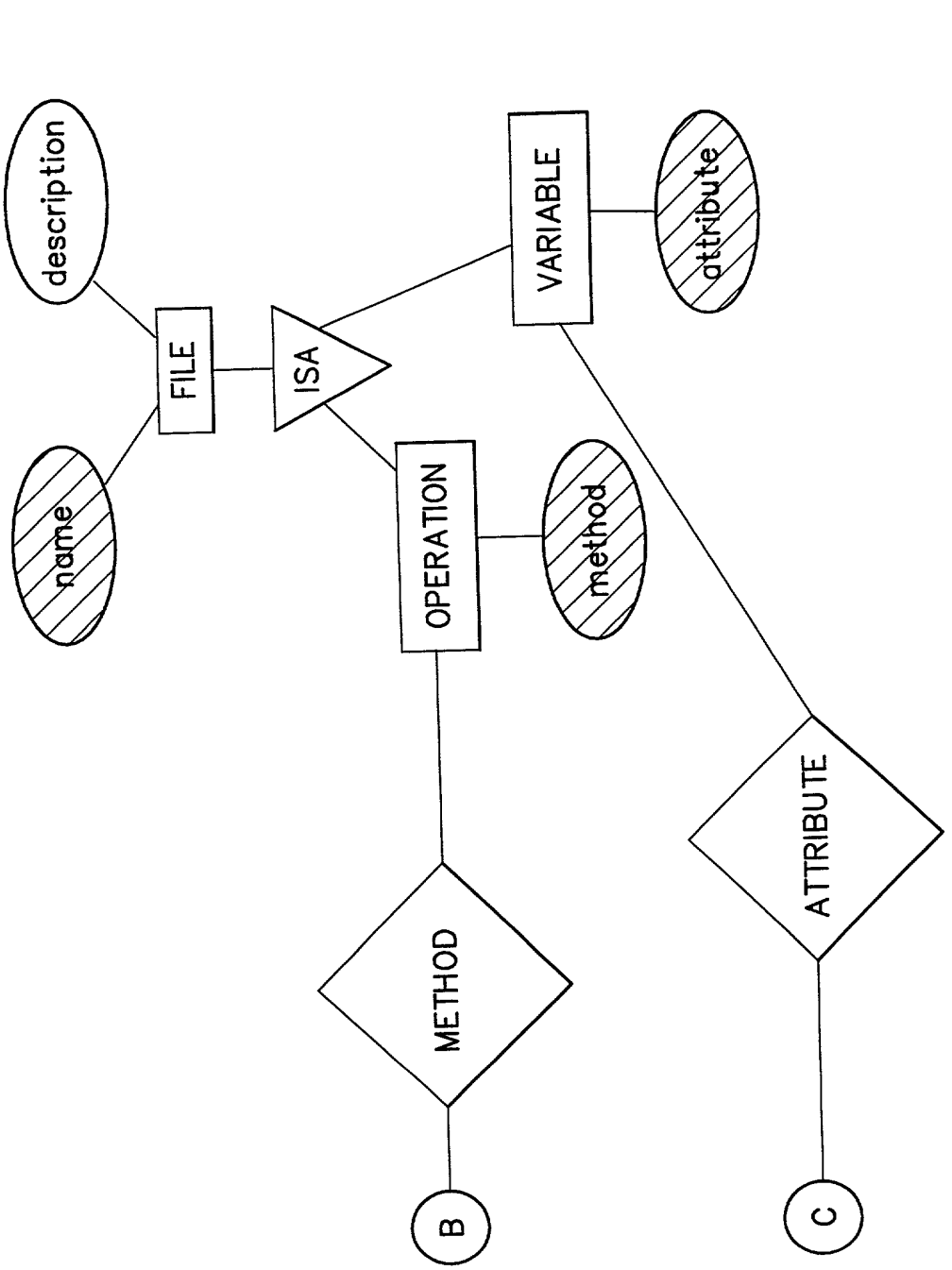

description
name
FILE
ISA
VARIABLE
attribute
OPERATION
method
METHOD
ATTRIBUTE
B
C
Source Code Information

FIG. 20C

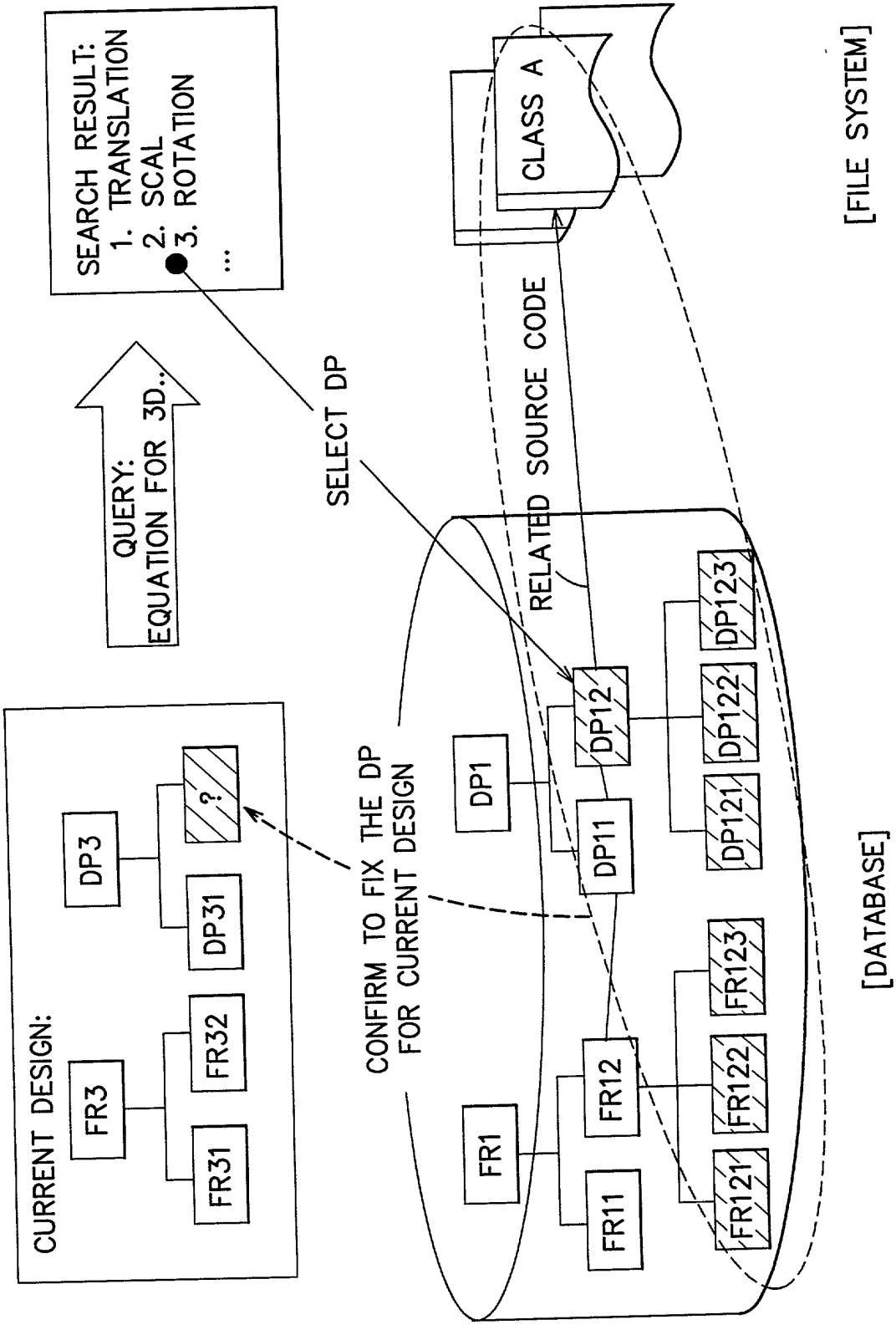

CURRENT DESIGN:
FR3
FR31
FR32
DP3
DP31
?
QUERY:
EQUATION FOR 3D..
SEARCH RESULT:
1. TRANSLATION
2. SCAL
3. ROTATION
...
SELECT DP
CONFIRM TO FIX THE DP
FOR CURRENT DESIGN
RELATED SOURCE CODE
CLASS A
[FILE SYSTEM]
[DATABASE]
FR1
FR11
FR12
FR121
FR122
FR123
DP1
DP11
DP12
DP121
DP122
DP123

FIG. 21

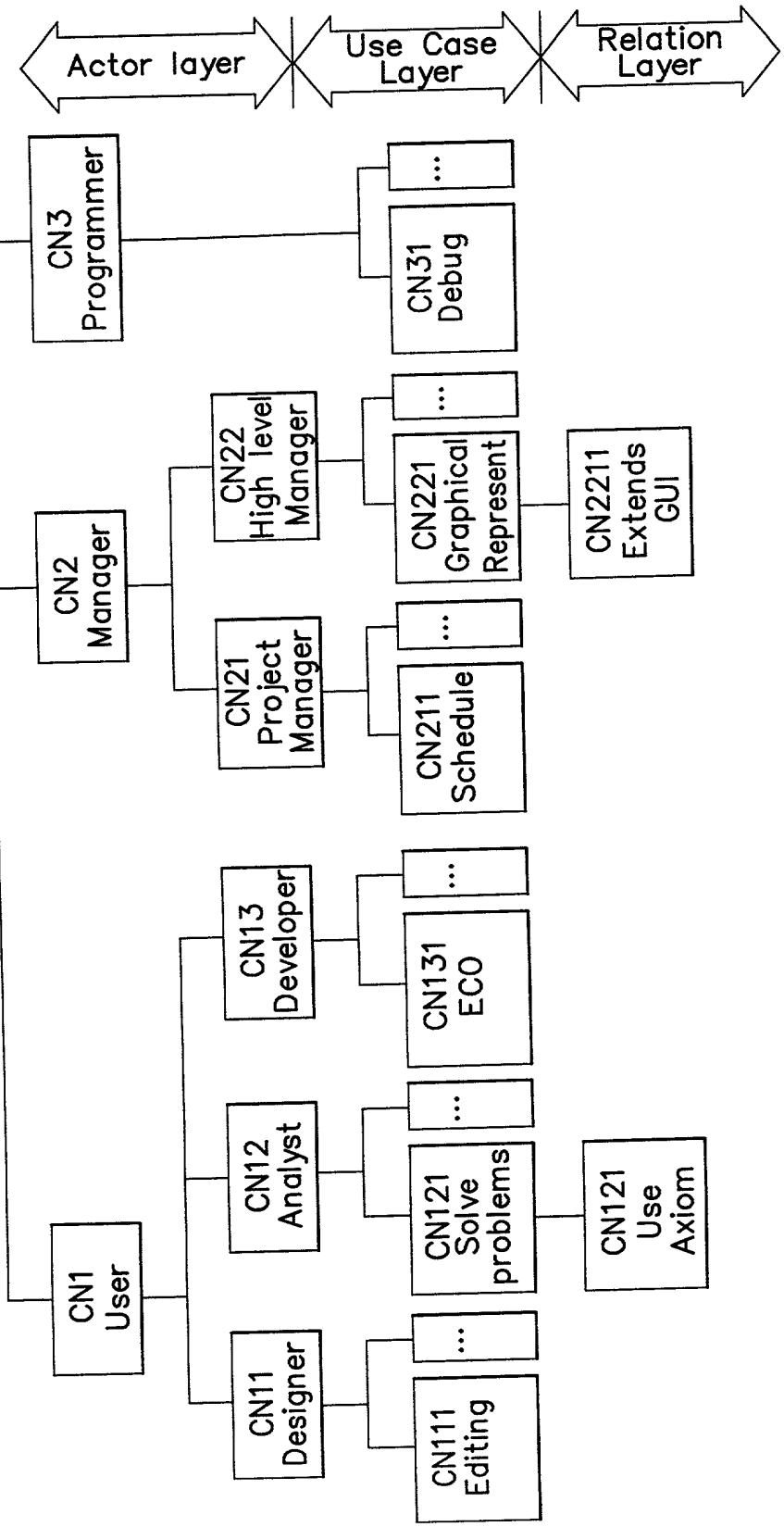

Actor layer
Use Case Layer
Relation Layer
CN1 User
CN2 Manager
CN3 Programmer
CN11 Designer
CN12 Analyst
CN13 Developer
CN21 Project Manager
CN22 High level Manager
CN31 Debug
CN111 Editing
CN121 Solve problems
CN121 Use Axiom
CN131 ECO
CN211 Schedule
CN221 Graphical Represent
CN2211 Extends GUI
...

FIG. 22

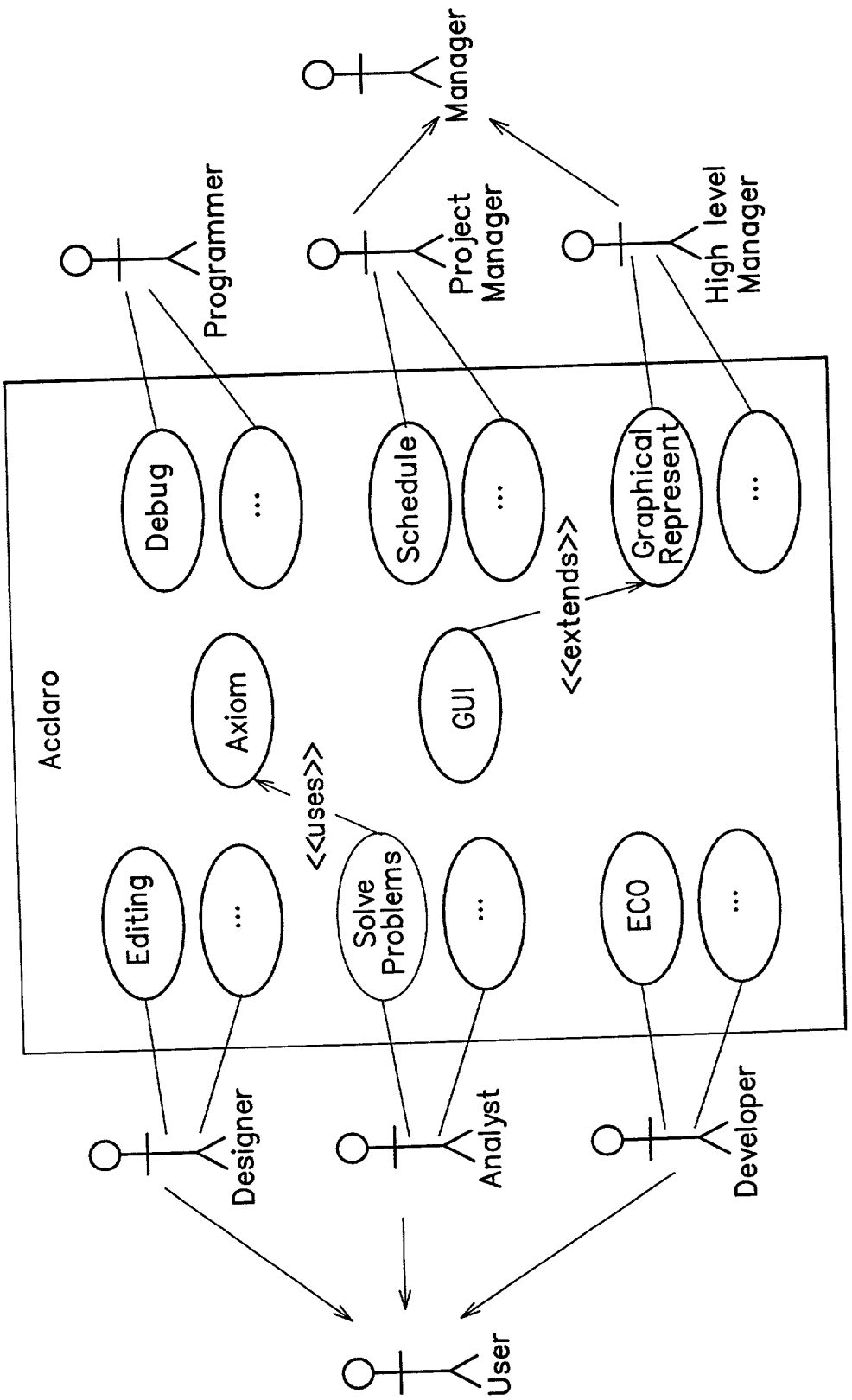

Manager
Programmer
Project Manager
High level Manager
Debug
...
Schedule
...
Graphical Represent
...
Acclaro
Axiom
GUI
<<uses>>
<<extends>>
Editing
...
Solve Problems
...
ECO
...
Designer
Analyst
Developer
User

FIG. 23

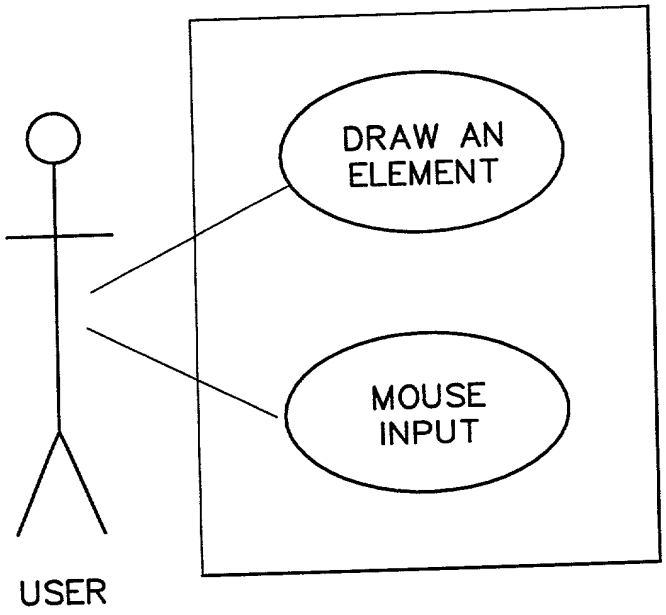


FIG. 24

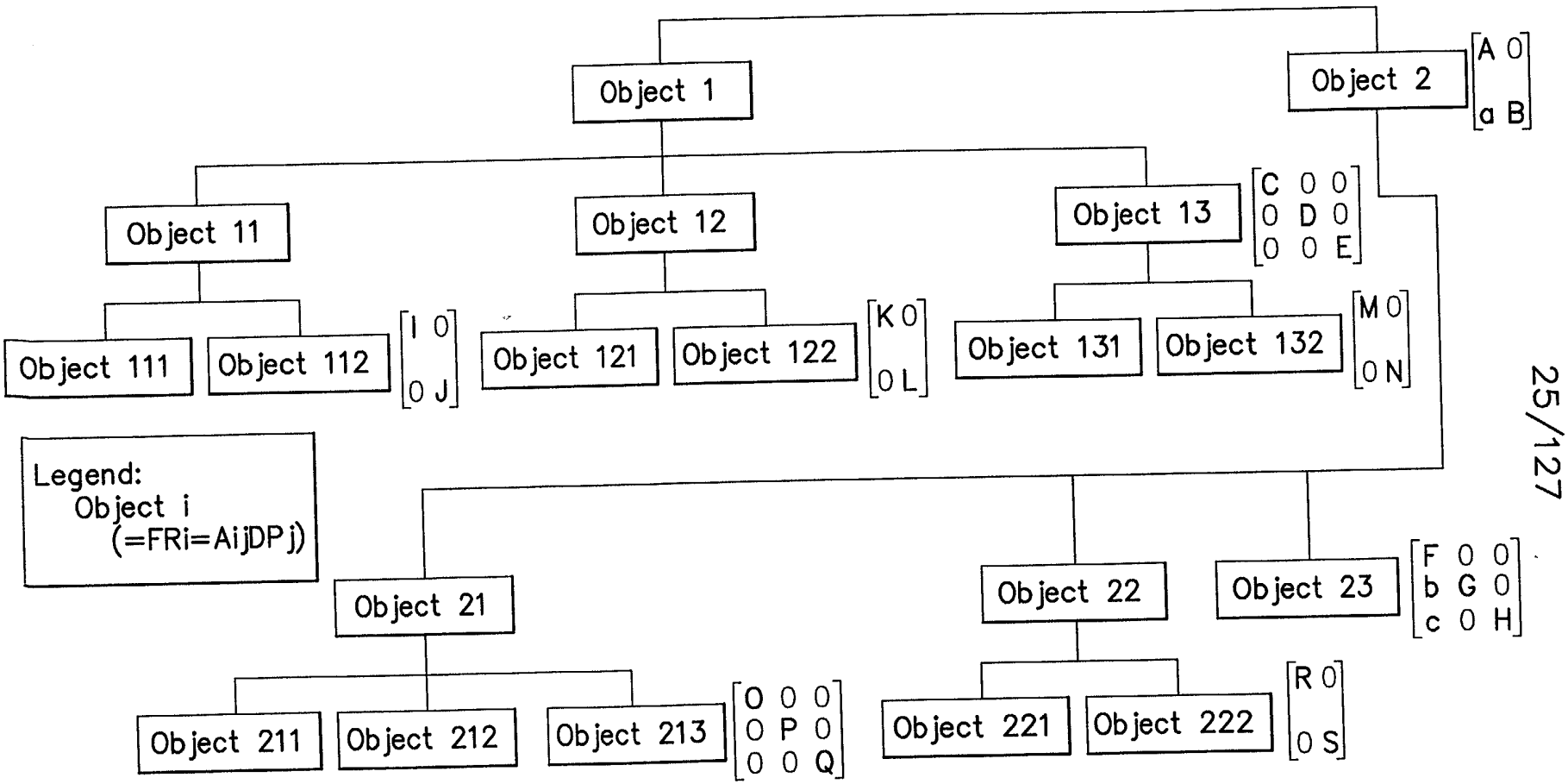


FIG. 25

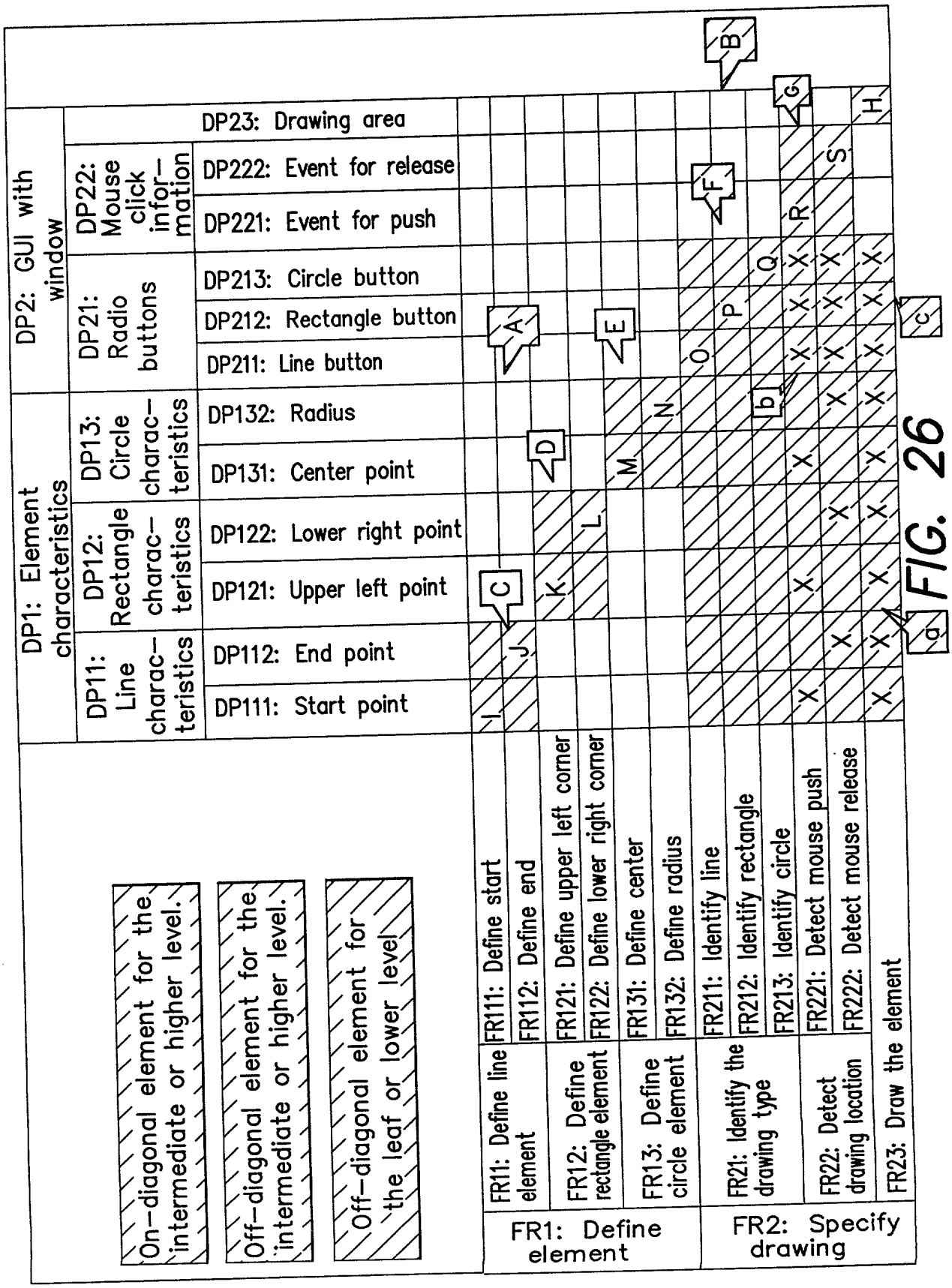

On-diagonal element for the intermediate or higher level.
Off-diagonal element for the intermediate or higher level.
Off-diagonal element for the leaf or lower level
DP1: Element characteristics
DP11: Line characteristics
DP12: Rectangle characteristics
DP13: Circle characteristics
DP2: GUI with window
DP21: Radio buttons
DP22: Mouse click information
DP23: Drawing area
DP111: Start point
DP112: End point
DP121: Upper left point
DP122: Lower right point
DP131: Center point
DP132: Radius
DP211: Line button
DP212: Rectangle button
DP213: Circle button
DP221: Event for push
DP222: Event for release
FR1: Define element
FR11: Define line element
FR111: Define start
FR112: Define end
FR12: Define rectangle element
FR121: Define upper left corner
FR122: Define lower right corner
FR13: Define circle element
FR131: Define center
FR132: Define radius
FR2: Specify drawing
FR21: Identify the drawing type
FR211: Identify line
FR212: Identify rectangle
FR213: Identify circle
FR22: Detect drawing location
FR221: Detect mouse push
FR222: Detect mouse release
FR23: Draw the element
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
a
b
c

FIG. 26

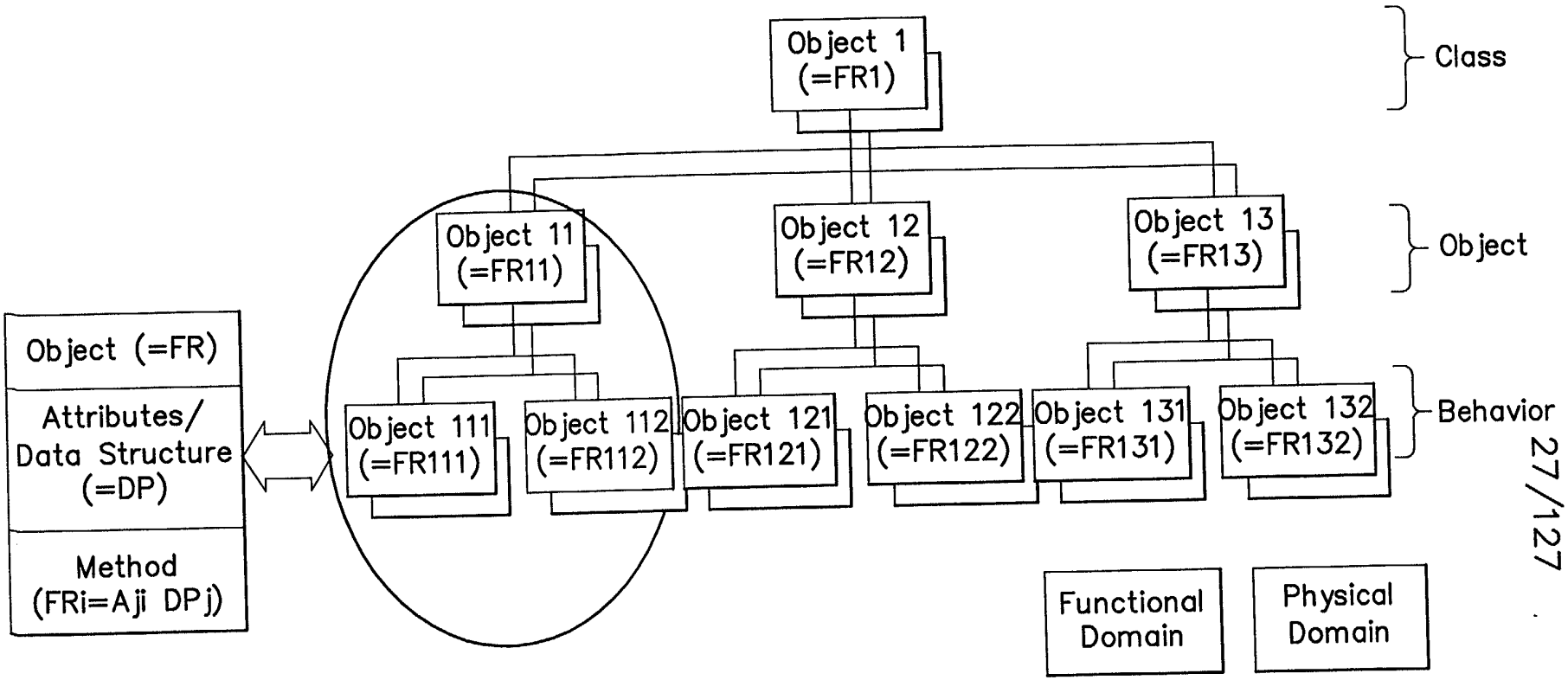
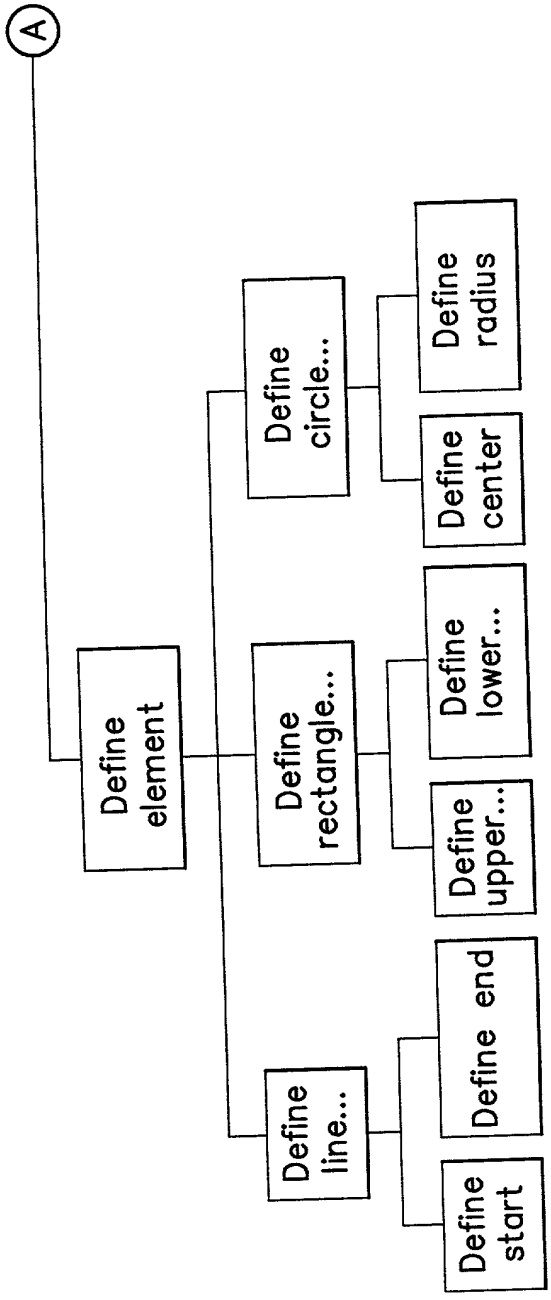


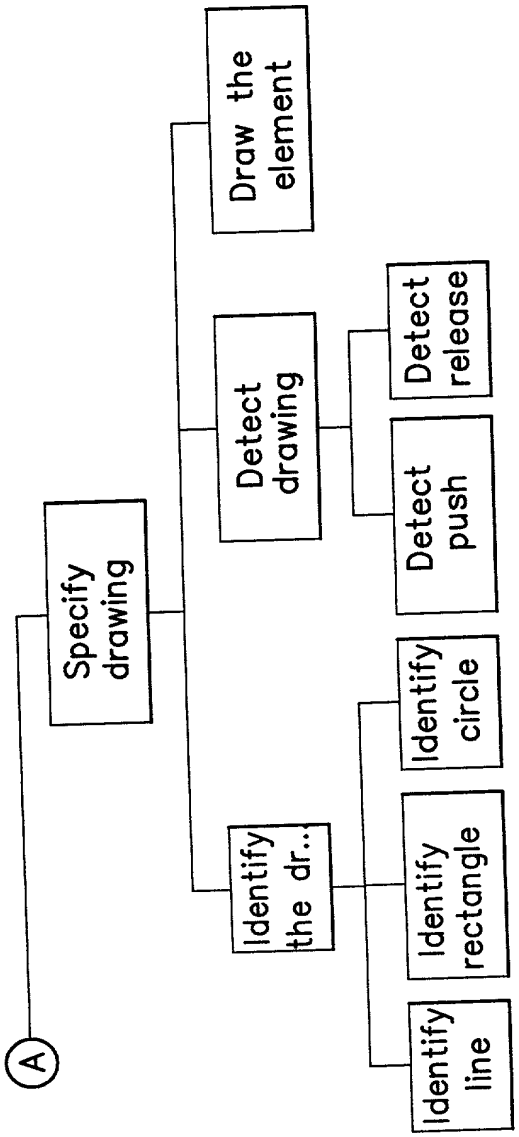
FIG. 27



(a) Functional Domain

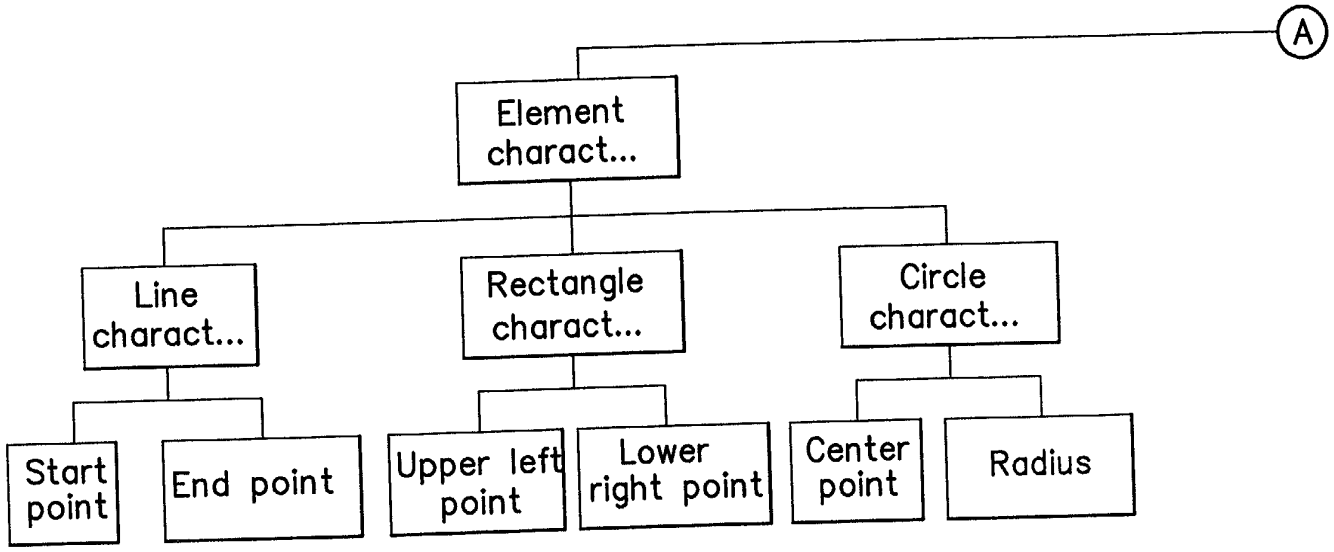
FIG. 28A(1) | FIG. 28A(2)

FIG. 28A(1)



(a) Functional Domain

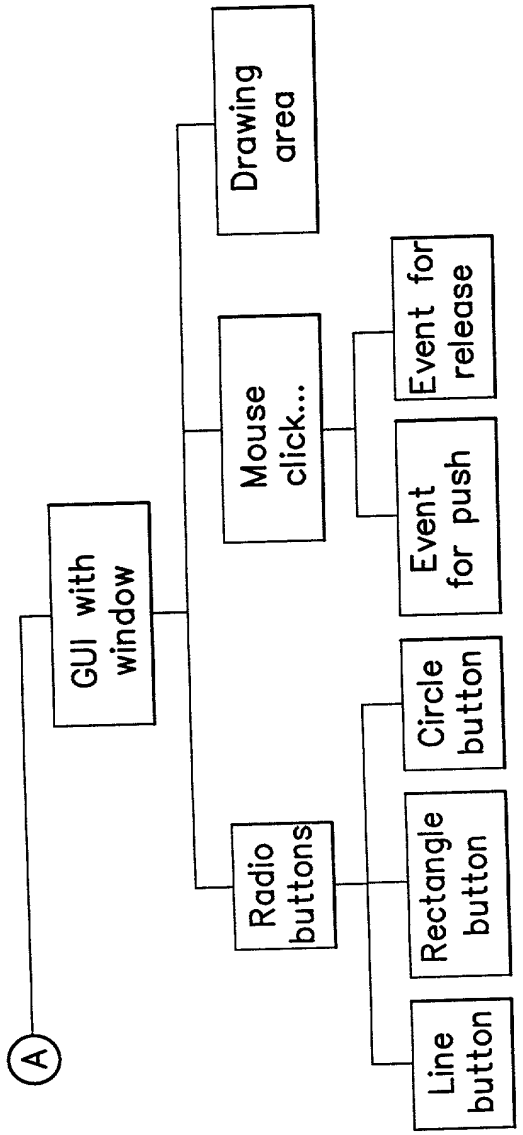
FIG. 28A(2)



(b) Physical Domain

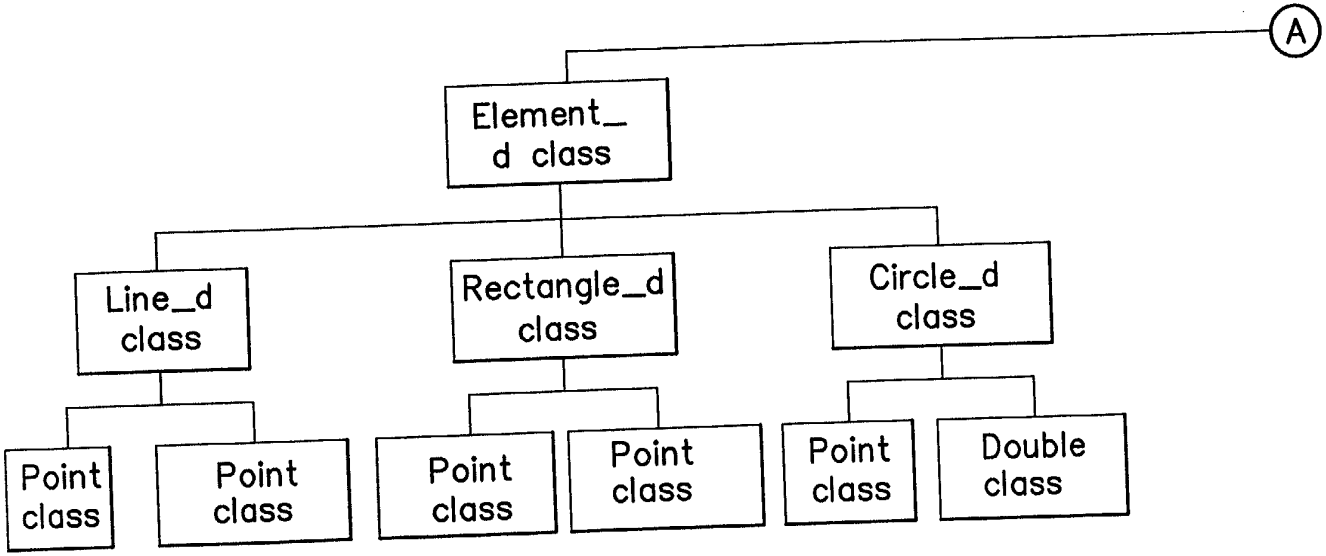
FIG. 28B(1)	FIG. 28B(2)

FIG. 28B(1)



(b) Physical Domain

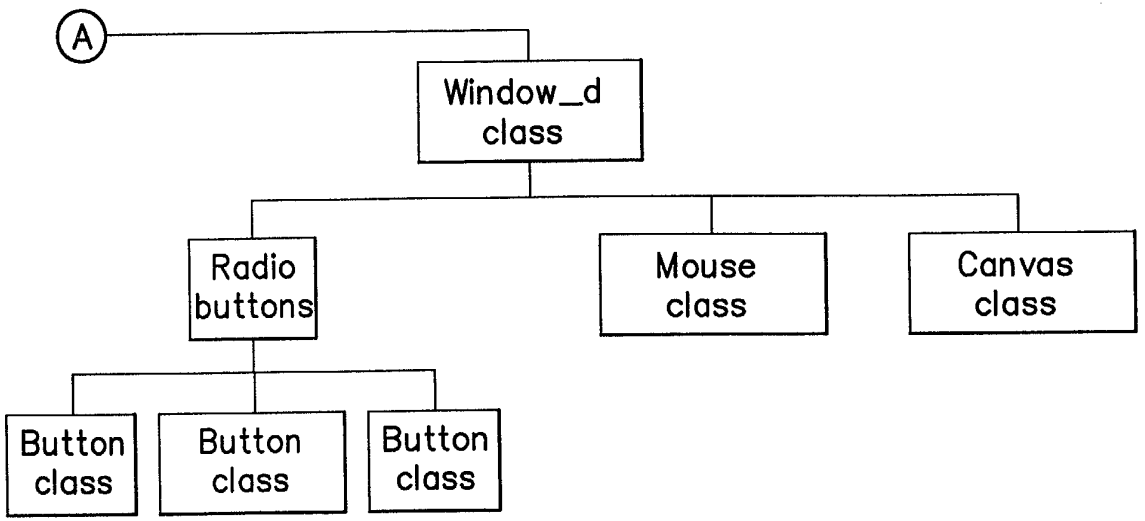
FIG. 28B(2)



(c) Process Domain

FIG. 28C(1)	FIG. 28C(2)

FIG. 28C(1)



(c) Process Domain

FIG. 28C(2)

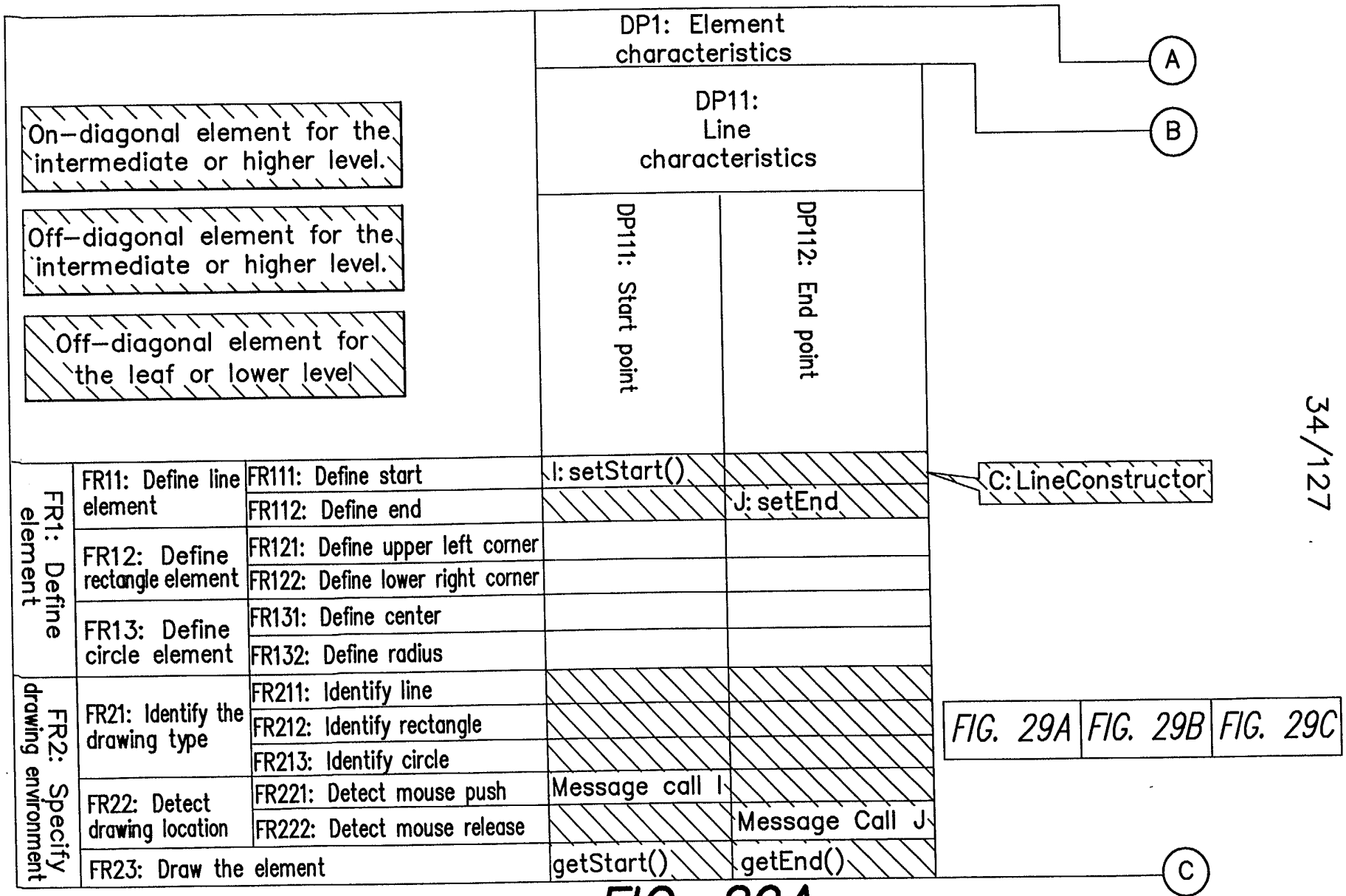

On-diagonal element for the intermediate or higher level.
Off-diagonal element for the intermediate or higher level.
Off-diagonal element for the leaf or lower level
DP1: Element characteristics
DP11: Line characteristics
DP111: Start point
DP112: End point
A
B
C
FR1: Define element
FR2: Specify drawing environment
FR11: Define line element
FR111: Define start
FR112: Define end
FR12: Define rectangle element
FR121: Define upper left corner
FR122: Define lower right corner
FR13: Define circle element
FR131: Define center
FR132: Define radius
FR21: Identify the drawing type
FR211: Identify line
FR212: Identify rectangle
FR213: Identify circle
FR22: Detect drawing location
FR221: Detect mouse push
FR222: Detect mouse release
FR23: Draw the element
I: setStart()
J: setEnd
C: LineConstructor
Message call I
Message Call J
getStart()
getEnd()
FIG. 29A
FIG. 29B
FIG. 29C

FIG. 29A

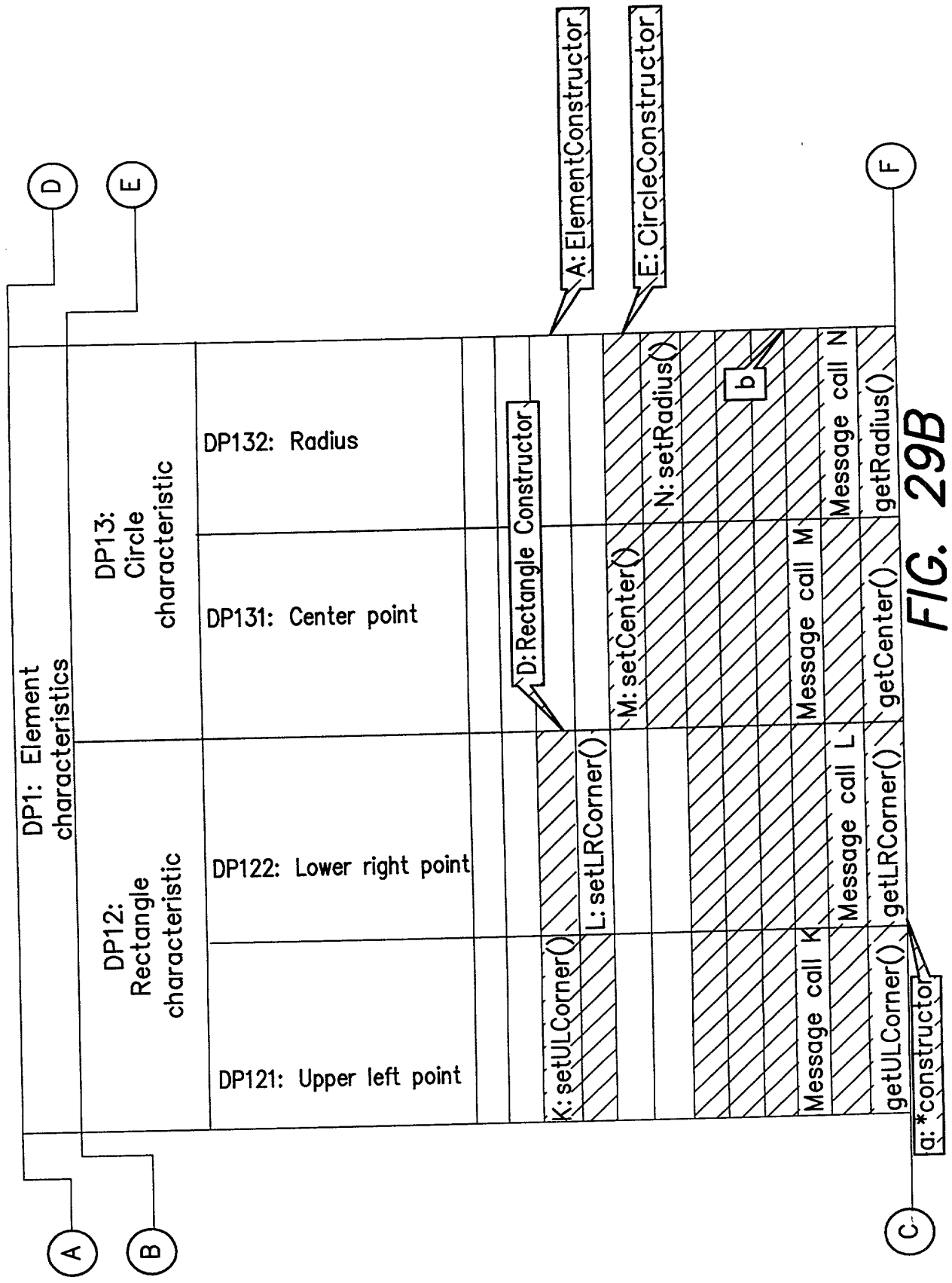

A: ElementConstructor
E: CircleConstructor
D
E
F
DP1: Element characteristics
DP13: Circle characteristic
DP12: Rectangle characteristic
DP132: Radius
DP131: Center point
DP122: Lower right point
DP121: Upper left point
D: Rectangle Constructor
K: setULCorner()
L: setLRCorner()
M: setCenter()
N: setRadius()
b
Message call K
Message call L
Message call M
Message call N
getULCorner()
getLRCorner()
getCenter()
getRadius()
a: *constructor
A
B
C

FIG. 29B

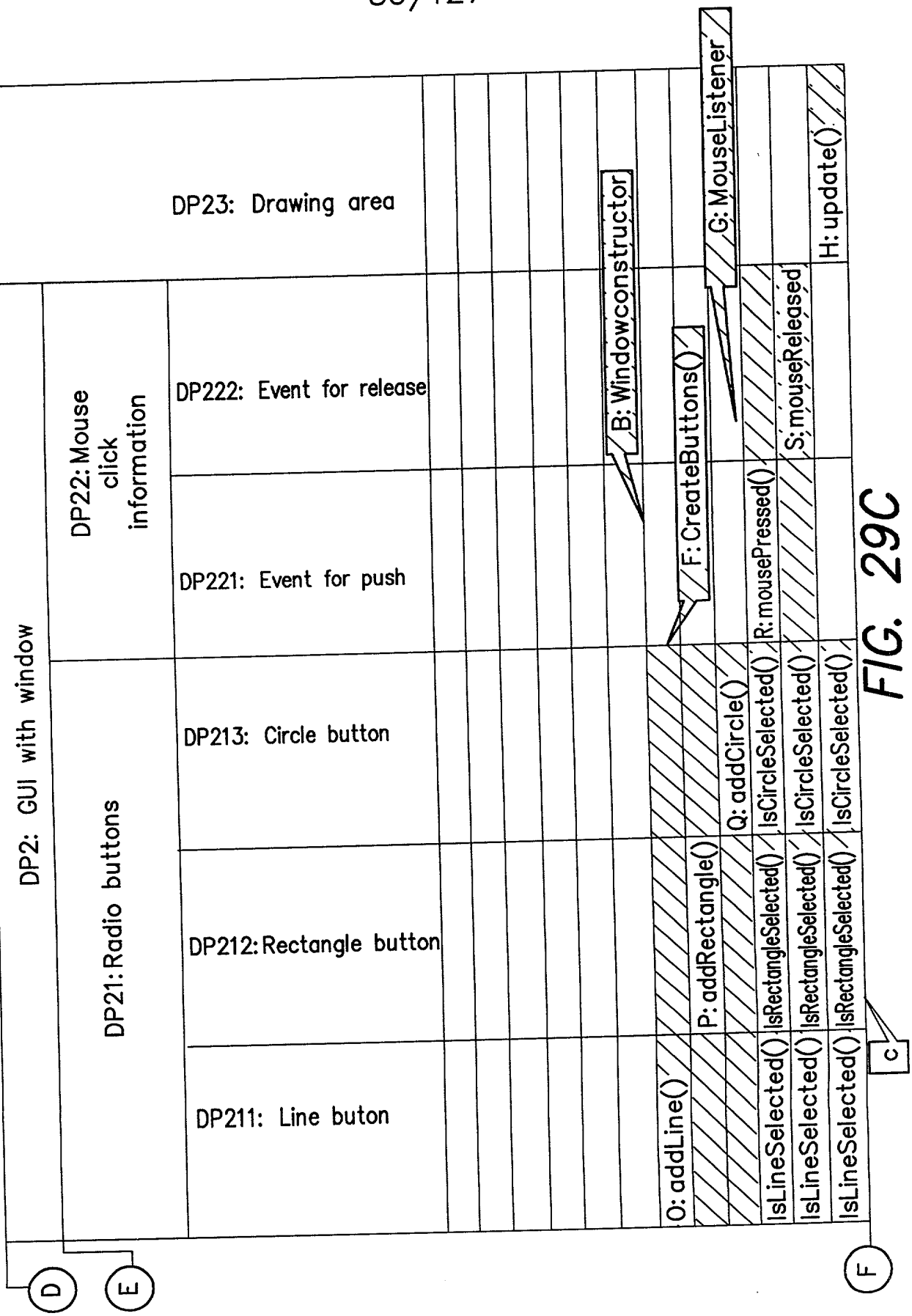

DP2: GUI with window
DP21: Radio buttons
DP22: Mouse click information
DP211: Line buton
DP212: Rectangle button
DP213: Circle button
DP221: Event for push
DP222: Event for release
DP23: Drawing area
O: addLine()
P: addRectangle()
Q: addCircle()
B: Windowconstructor
F: CreateButtons()
G: MouseListener
R: mousePressed()
S: mouseReleased
H: update()
IsLineSelected()
IsRectangleSelected()
IsCircleSelected()
D
E
F
C

FIG. 29C

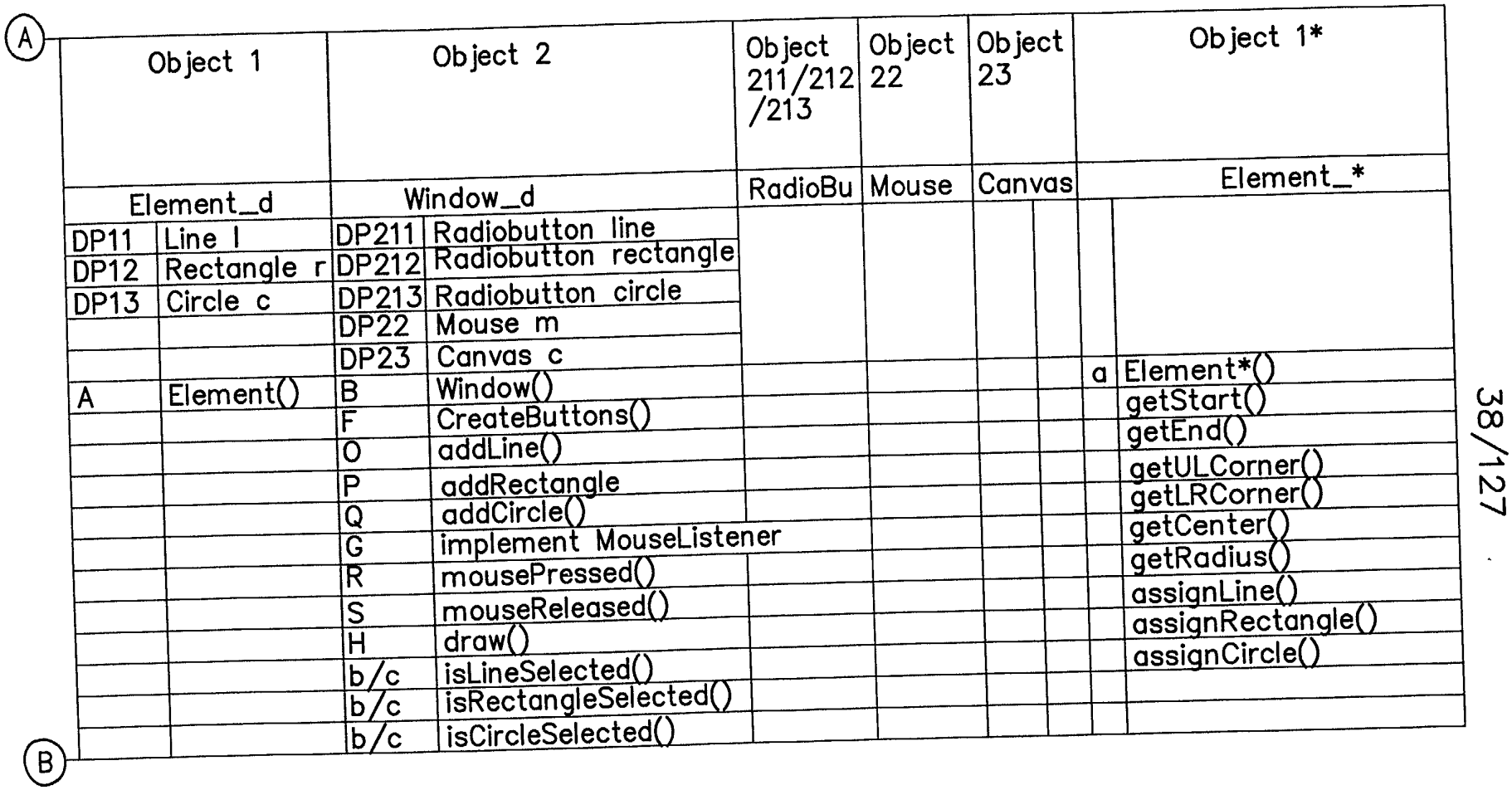
Object	Object 111/112/121/122/131	Object 132	Object 11		Object 12		Object 13	
Name	Point	Double	Line_d		Rectangle_d		Circle_d	
Attribute			DP111	Point start	DP121	Point upper_left	DP131	Point center
			DP112	Point end	DP122	Point lower_right	DP132	Double radius
Method			C	Line()	D	Rectangle()	E	Center()
			I	setStart()	K	setULCorner()	M	setCenter()
			J	setEnd()	L	setLRCorner()	N	setRadius()

A

B

FIG. 30A | FIG. 30B

FIG. 30A



Object 1		Object 2		Object 211/212/213	Object 22	Object 23		Object 1*	
Element_d		Window_d		RadioBu	Mouse	Canvas		Element_*	
DP11	Line l	DP211	Radiobutton line						
DP12	Rectangle r	DP212	Radiobutton rectangle						
DP13	Circle c	DP213	Radiobutton circle						
		DP22	Mouse m						
		DP23	Canvas c						
A	Element()	B	Window()					a	Element*()
		F	CreateButtons()						getStart()
		O	addLine()						getEnd()
		P	addRectangle						getULCorner()
		Q	addCircle()						getLRCorner()
		G	implement MouseListener						getCenter()
		R	mousePressed()						getRadius()
		S	mouseReleased()						assignLine()
		H	draw()						assignRectangle()
		b/c	isLineSelected()						assignCircle()
		b/c	isRectangleSelected()						
		b/c	isCircleSelected()						

FIG. 30B

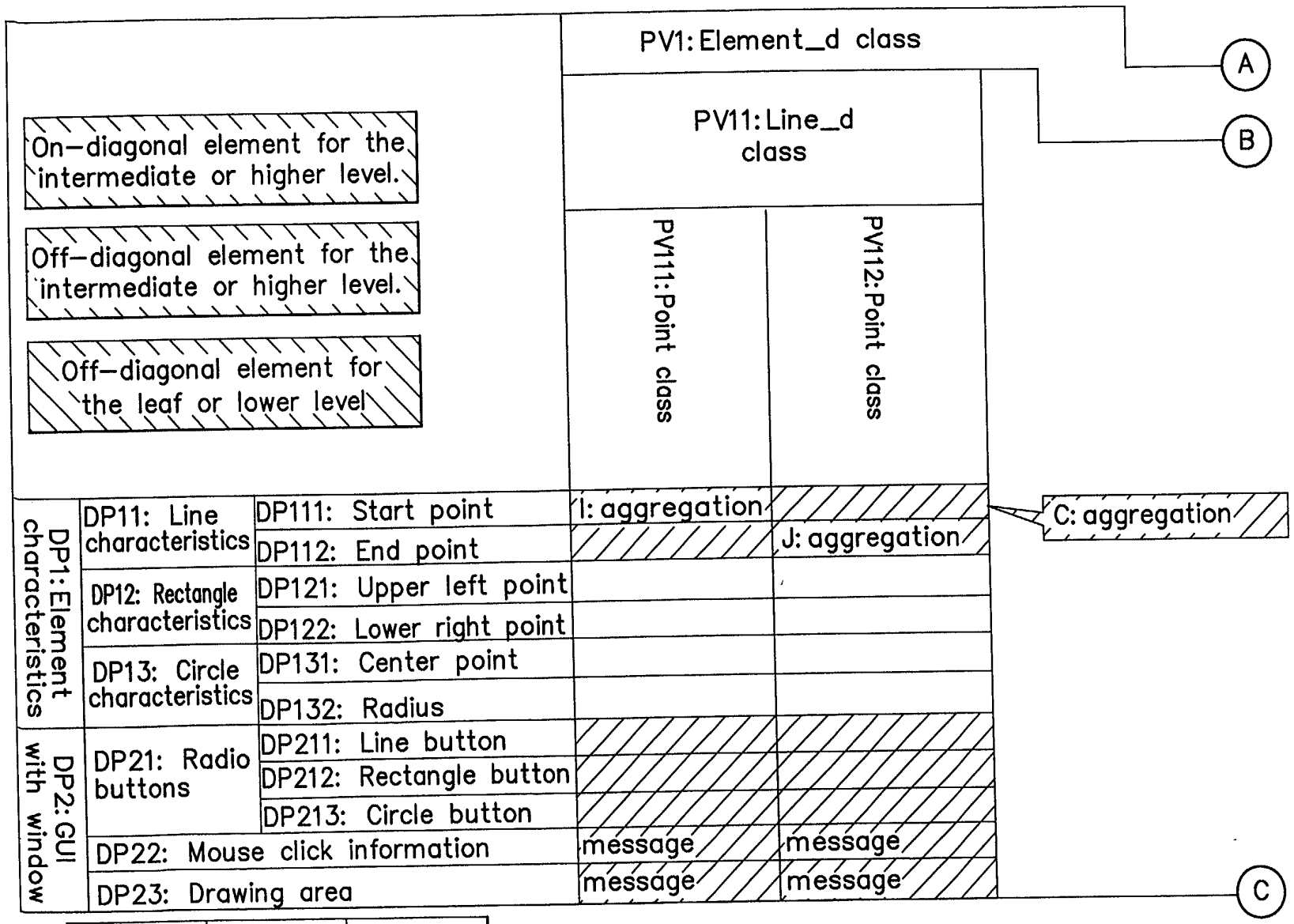


FIG. 31A | FIG. 31B | FIG. 31C

FIG. 31A

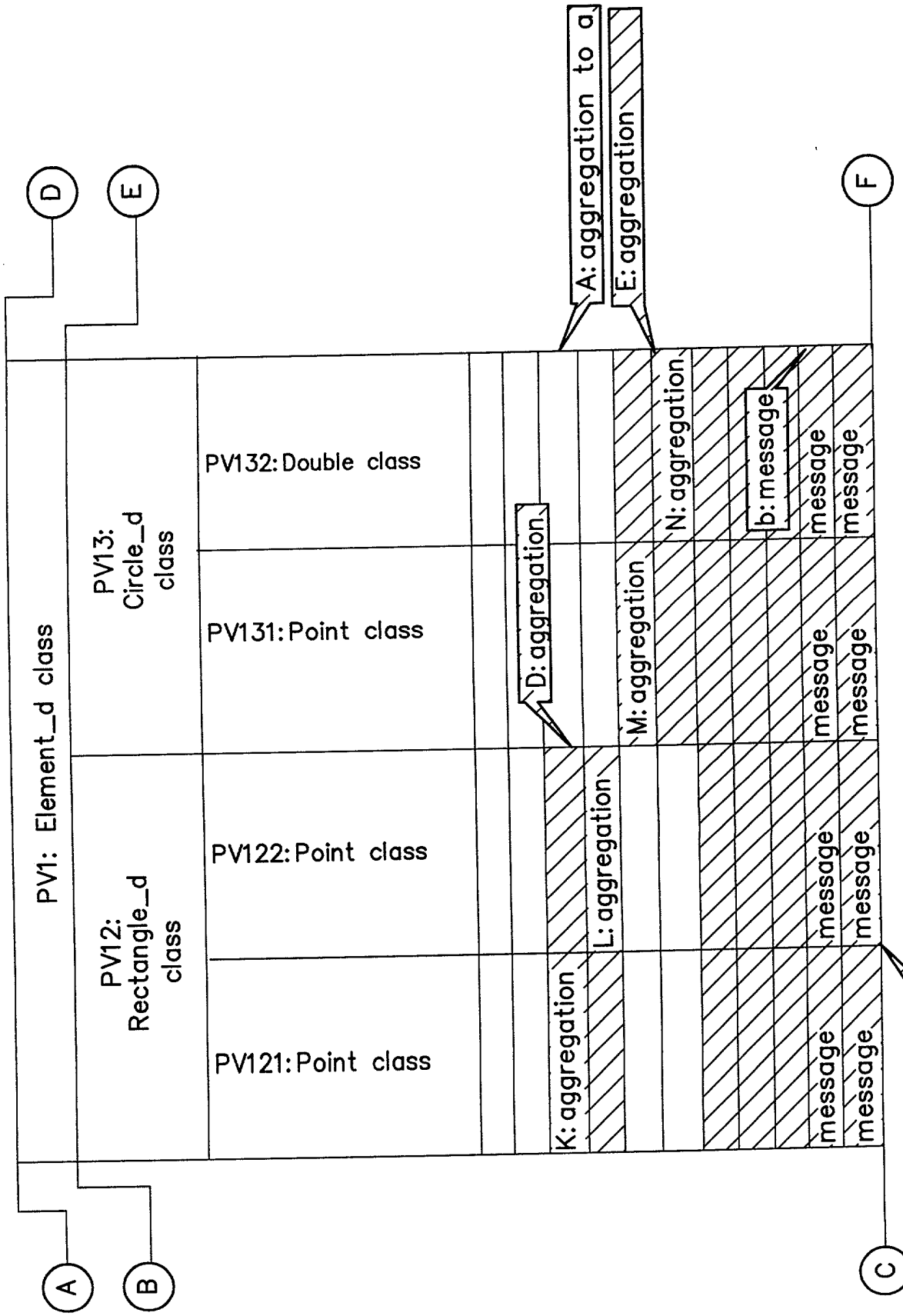

A
B
C
D
E
F
PV1: Element_d class
PV13: Circle_d class
PV12: Rectangle_d class
PV132: Double class
PV131: Point class
PV122: Point class
PV121: Point class
A: aggregation to a
E: aggregation
D: aggregation
N: aggregation
M: aggregation
L: aggregation
K: aggregation
b: message
message
a: aggregation to object 2

FIG. 31B

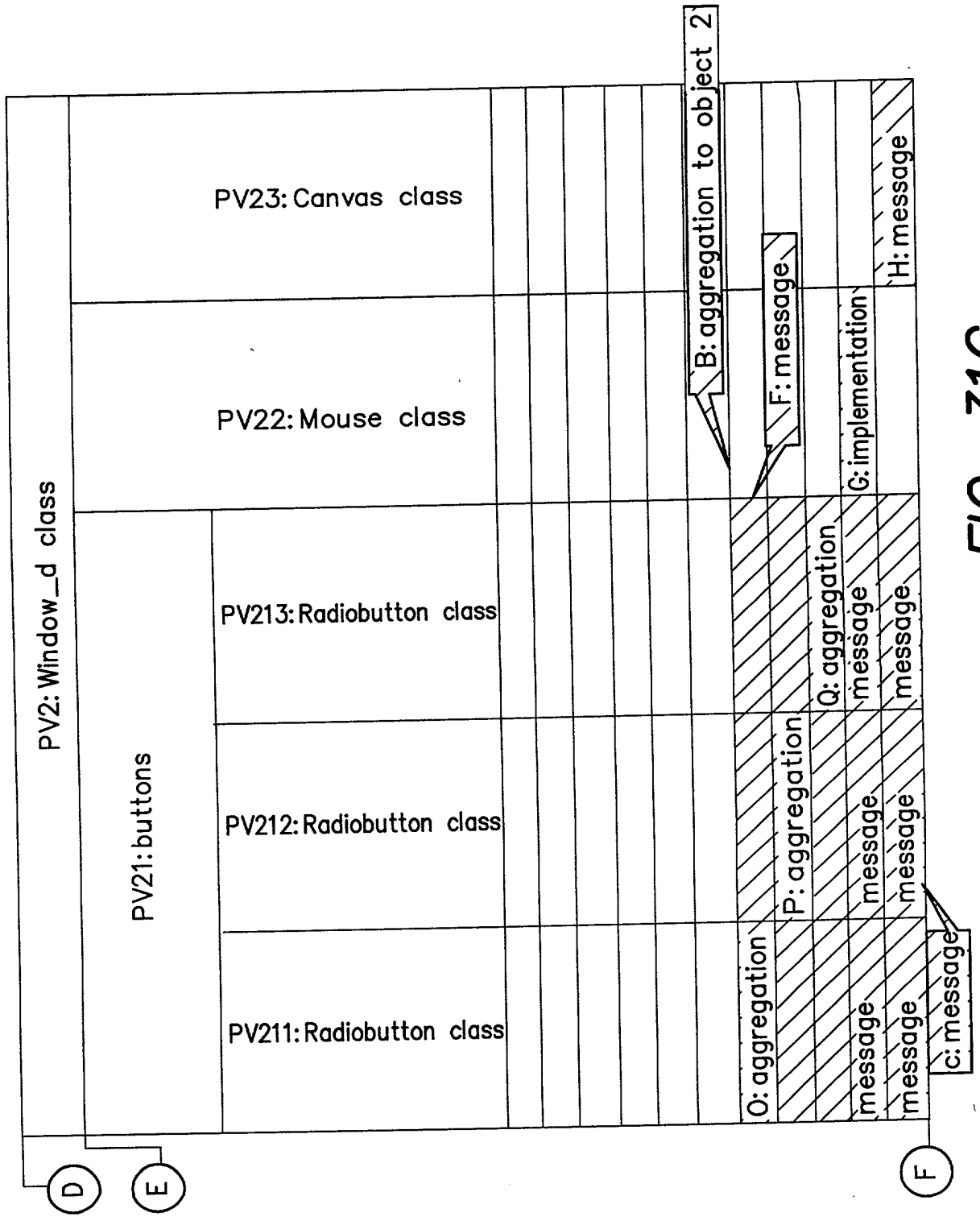

PV2: Window_d class
PV21: buttons
PV23: Canvas class
PV22: Mouse class
PV213: Radiobutton class
PV212: Radiobutton class
PV211: Radiobutton class
B: aggregation to object 2
F: message
G: implementation
H: message
Q: aggregation
message
message
message
P: aggregation
message
message
message
O: aggregation
message
message
message
c: message
D
E
F

FIG. 31C

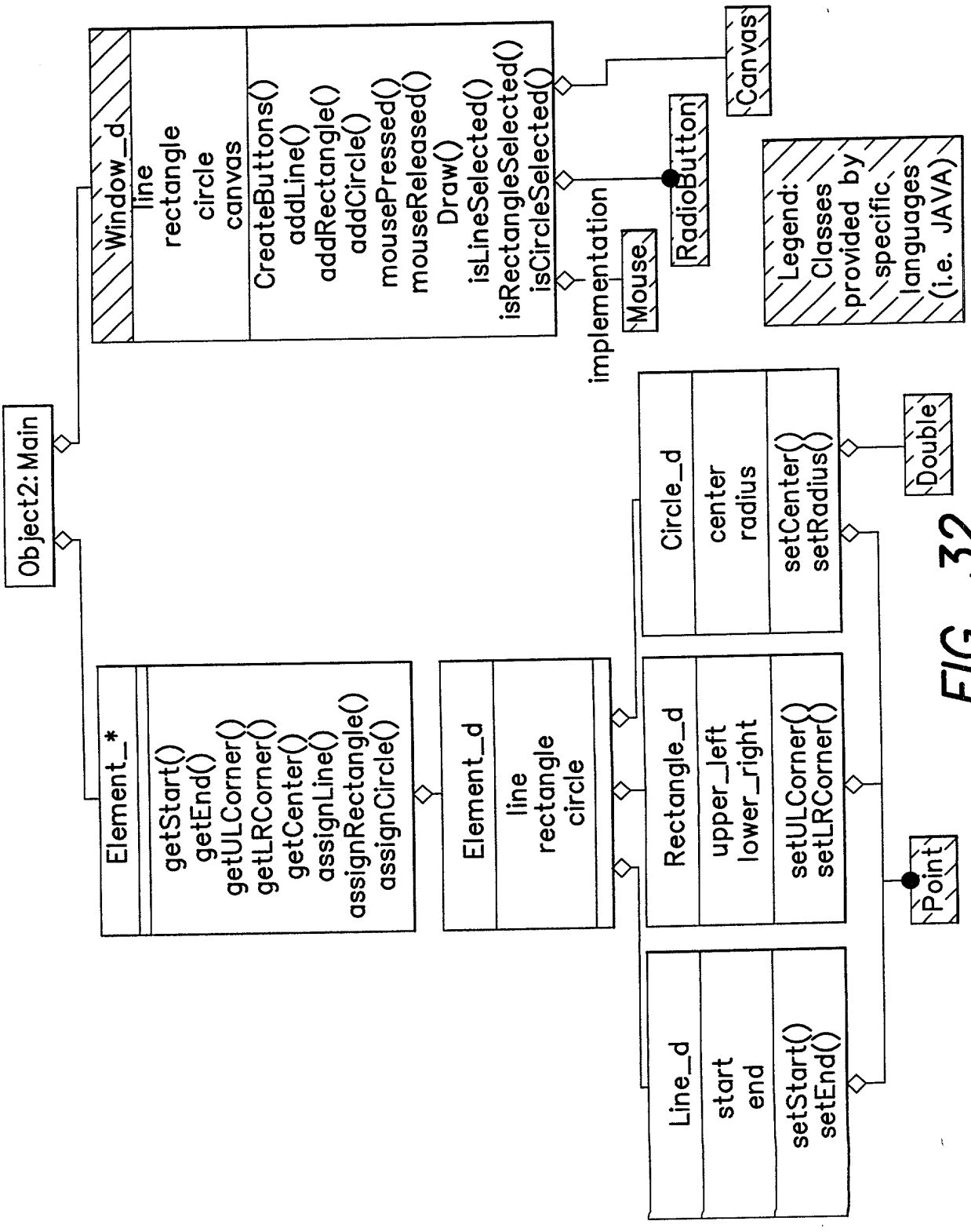

Object2: Main
Window_d
line
rectangle
circle
canvas
CreateButtons()
addLine()
addRectangle()
addCircle()
mousePressed()
mouseReleased()
Draw()
isLineSelected()
isRectangleSelected()
isCircleSelected()
implementation
Mouse
RadioButton
Canvas
Legend:
Classes
provided by
specific
languages
(i.e. JAVA)
Element_*
getStart()
getEnd()
getULCorner()
getLRCorner()
getCenter()
assignLine()
assignRectangle()
assignCircle()
Element_d
line
rectangle
circle
Circle_d
center
radius
setCenter()
setRadius()
Double
Rectangle_d
upper_left
lower_right
setULCorner()
setLRCorner()
Point
Line_d
start
end
setStart()
setEnd()

FIG. 32

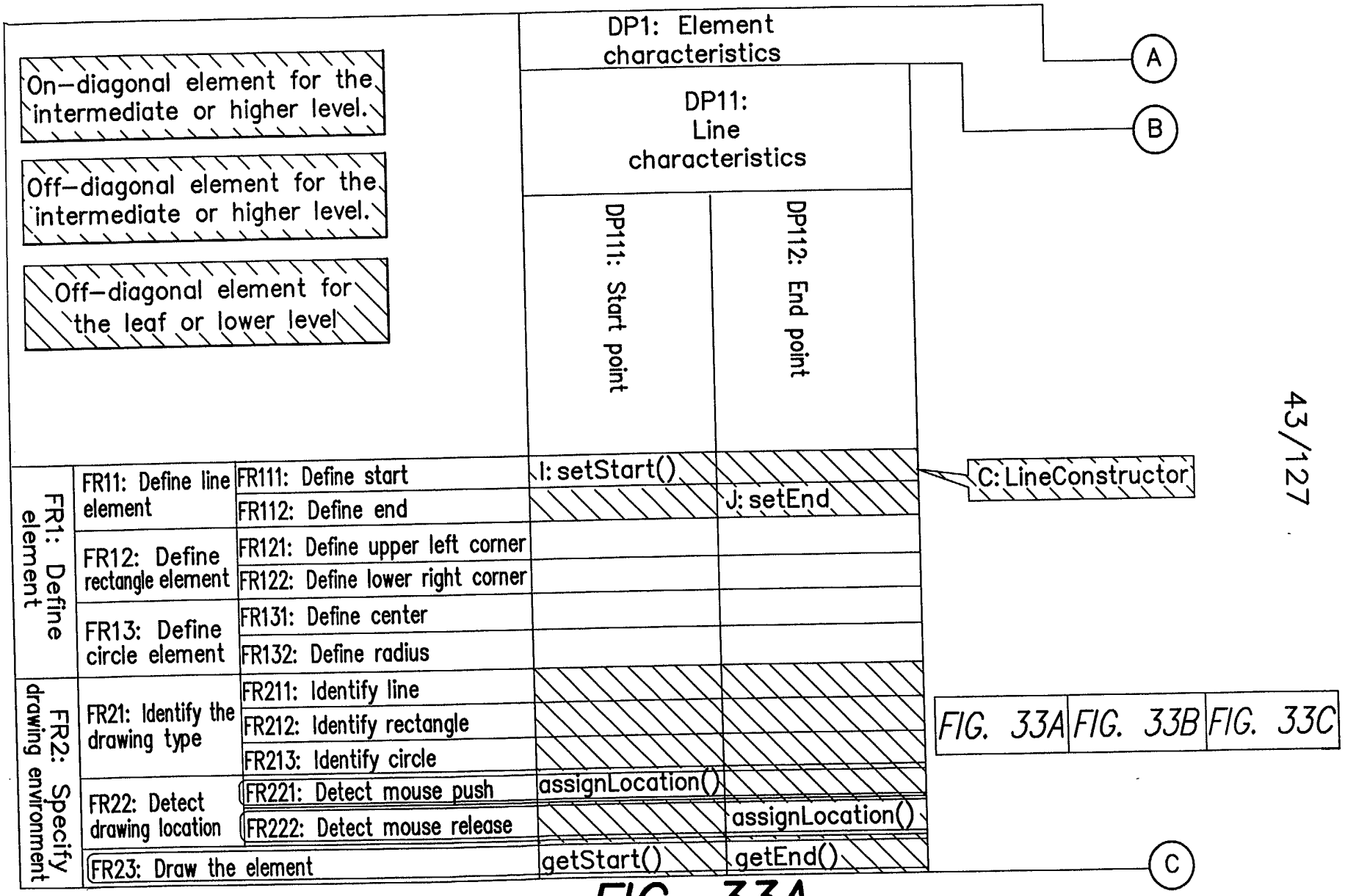


FIG. 33A | FIG. 33B | FIG. 33C

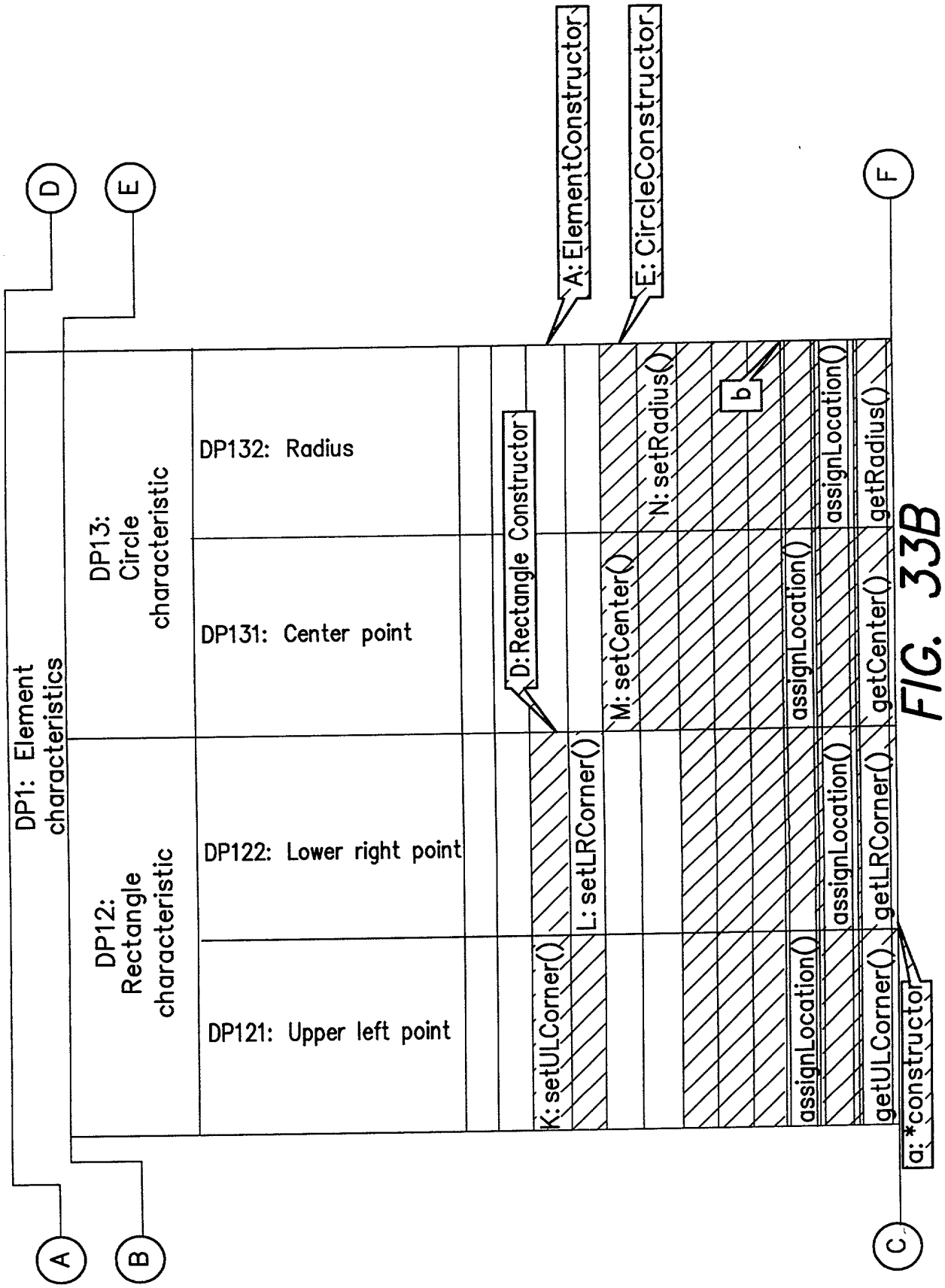

A: ElementConstructor
E: CircleConstructor
D
E
F
DP1: Element characteristics
DP13: Circle characteristic
DP12: Rectangle characteristic
DP132: Radius
DP131: Center point
DP122: Lower right point
DP121: Upper left point
D: Rectangle Constructor
K: setULCorner()
L: setLRCorner()
M: setCenter()
N: setRadius()
b
assignLocation()
assignLocation()
getCenter()
getRadius()
assignLocation()
assignLocation()
getULCorner()
getLRCorner()
a: *constructor
A
B
C

FIG. 33B

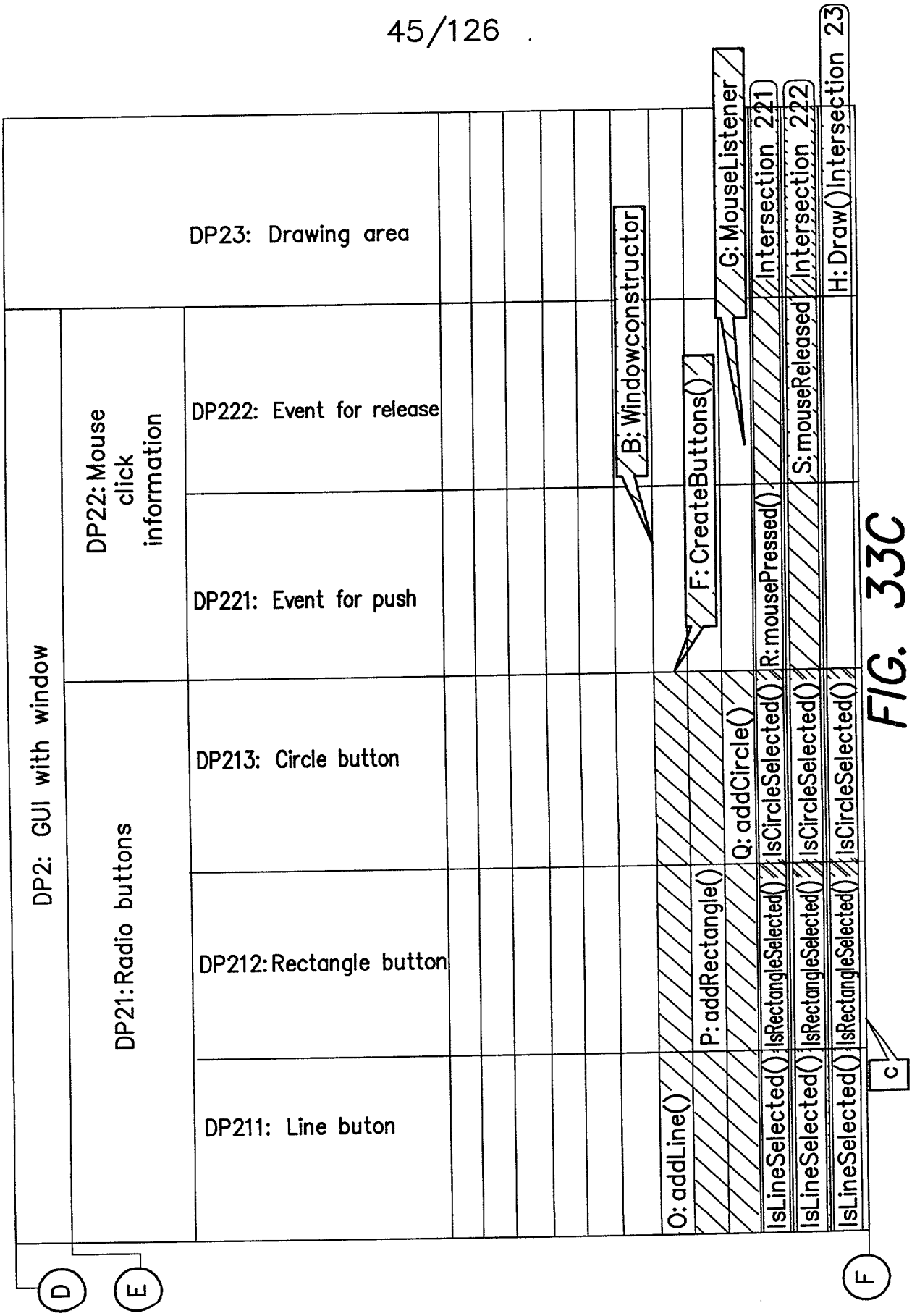

DP2: GUI with window
DP21: Radio buttons
DP22: Mouse click information
DP211: Line buton
DP212: Rectangle button
DP213: Circle button
DP221: Event for push
DP222: Event for release
DP23: Drawing area
O: addLine()
P: addRectangle()
Q: addCircle()
B: Windowconstructor
F: CreateButtons()
G: MouseListener
R: mousePressed()
S: mouseReleased
H: Draw()Intersection 23
Intersection 221
Intersection 222
IsLineSelected()
IsRectangleSelected()
IsCircleSelected()
D
E
F
C

FIG. 33C

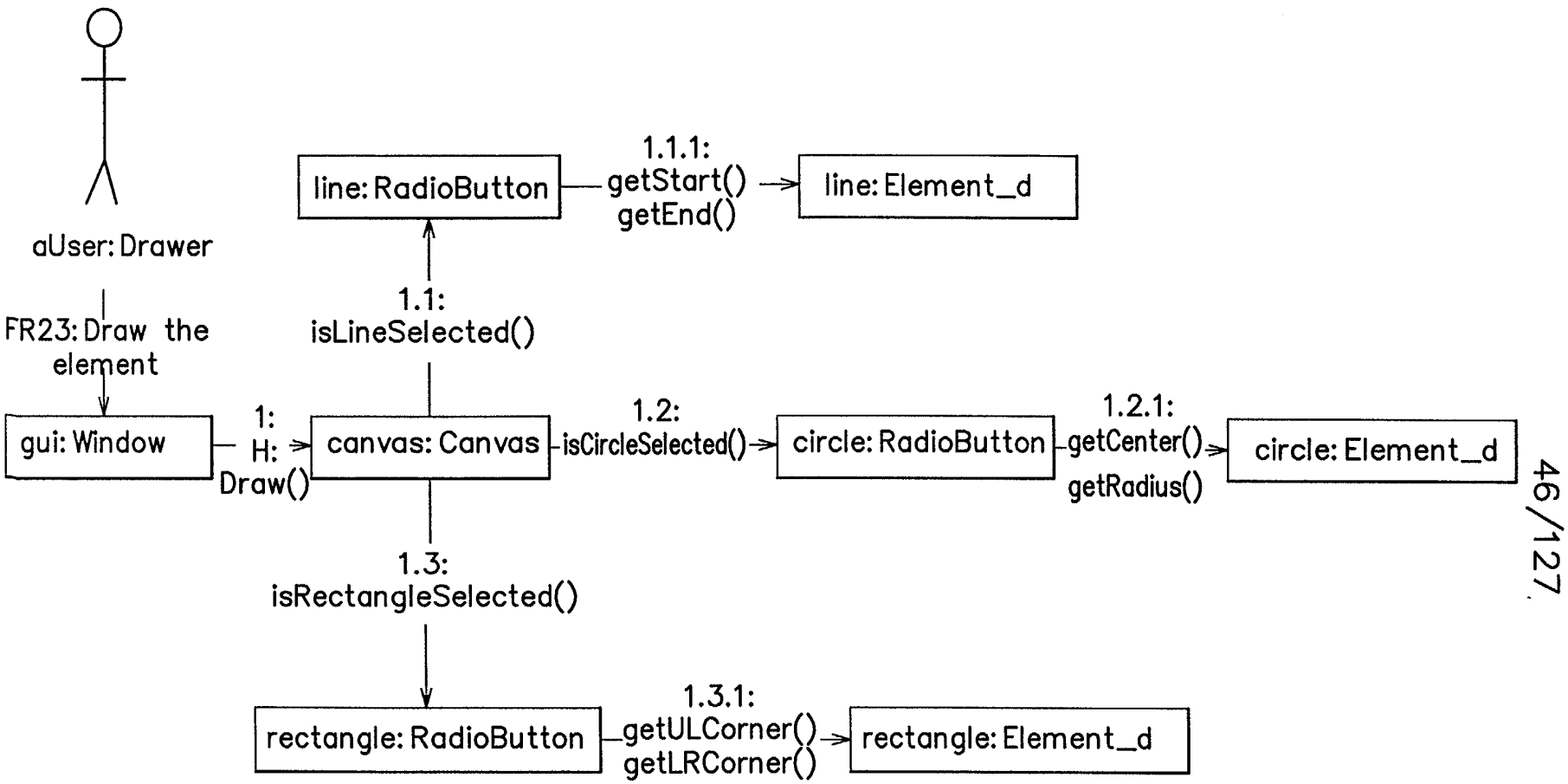


FIG. 34

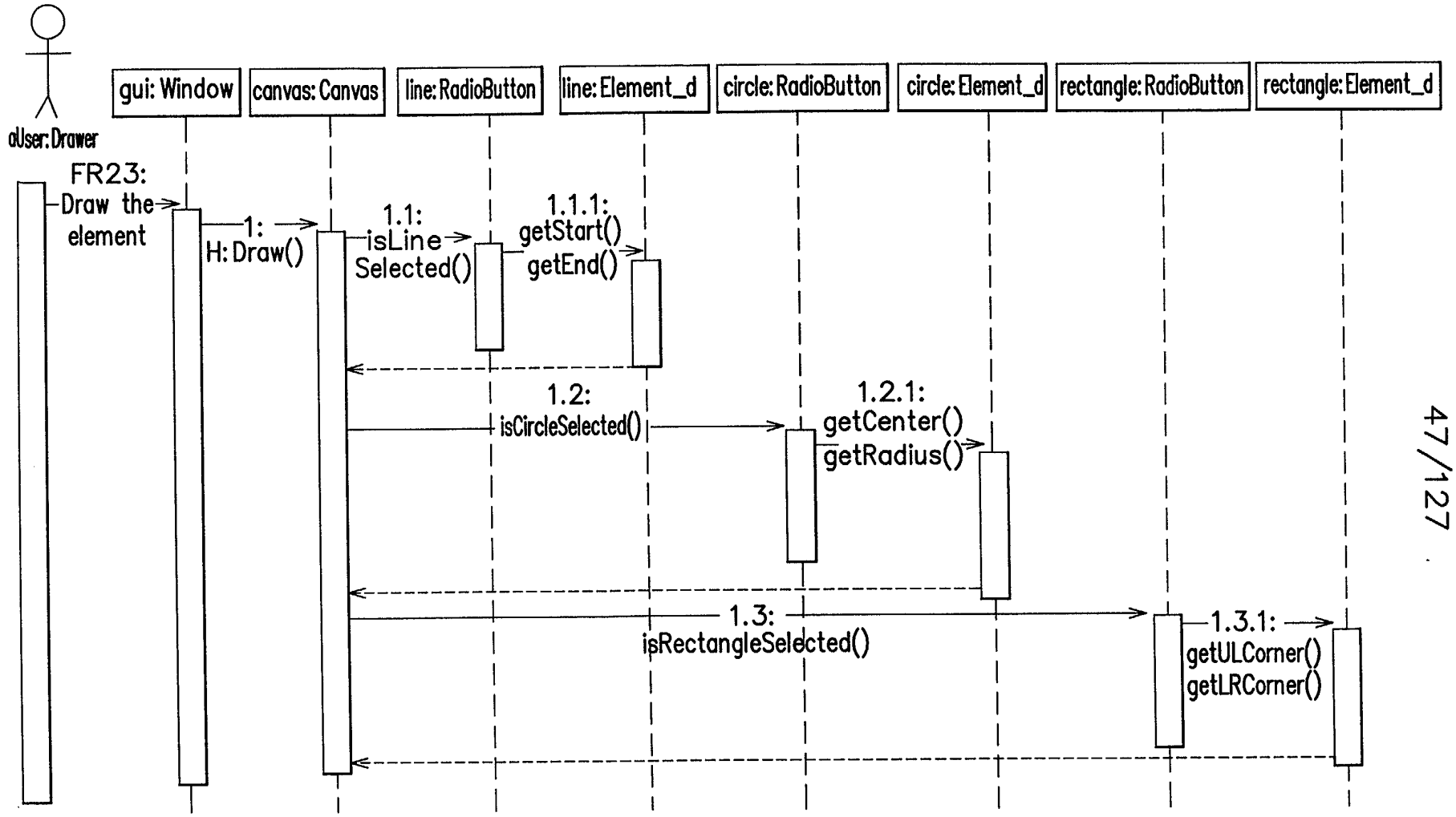

aUser:Drawer
gui: Window
canvas: Canvas
line: RadioButton
line: Element_d
circle: RadioButton
circle: Element_d
rectangle: RadioButton
rectangle: Element_d
FR23:
Draw the
element
1:
H: Draw()
1.1:
isLine
Selected()
1.1.1:
getStart()
getEnd()
1.2:
isCircleSelected()
1.2.1:
getCenter()
getRadius()
1.3:
isRectangleSelected()
1.3.1:
getULCorner()
getLRCorner()

FIG. 35

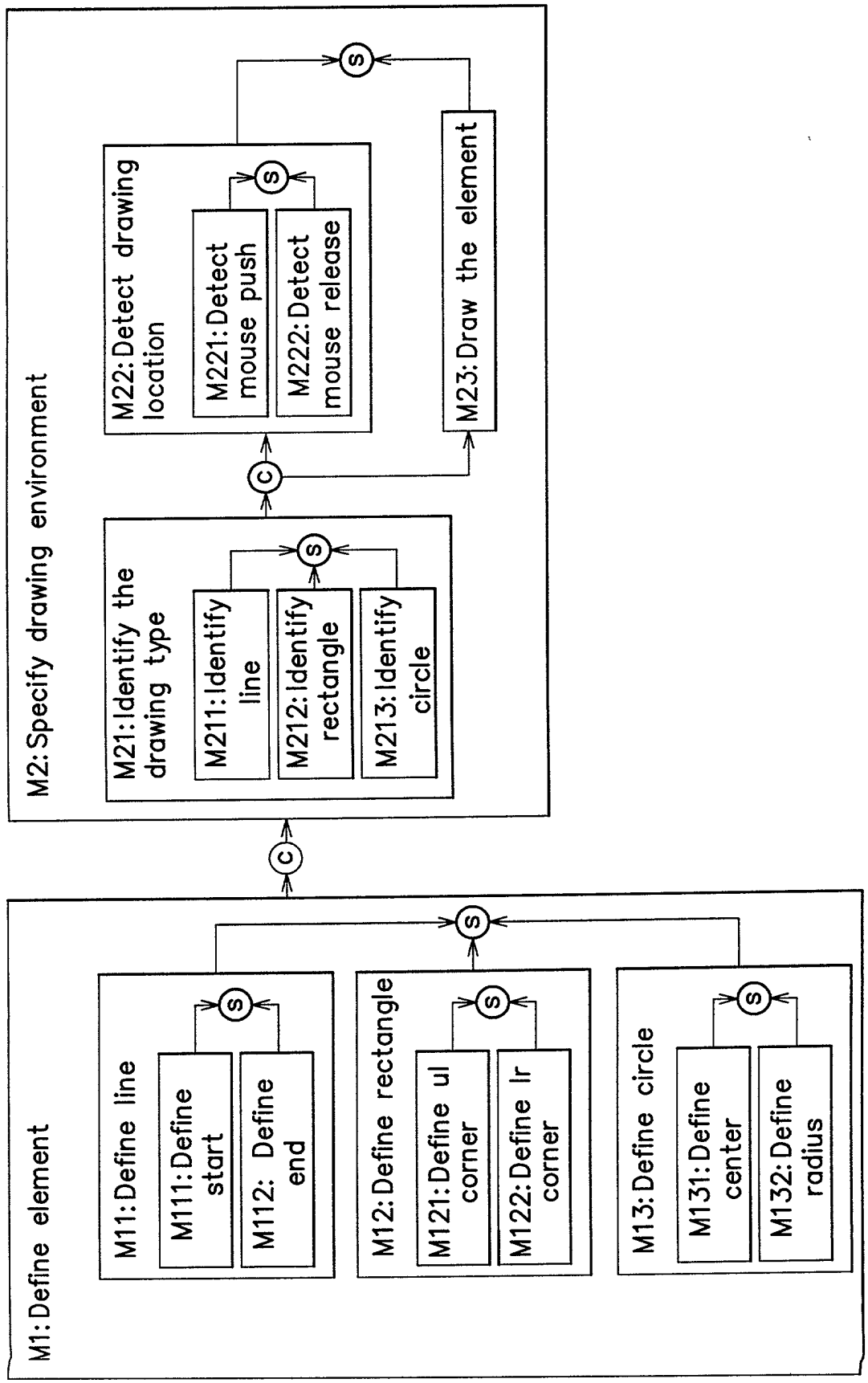

M1: Define element
M11: Define line
M111: Define start
M112: Define end
M12: Define rectangle
M121: Define ul corner
M122: Define lr corner
M13: Define circle
M131: Define center
M132: Define radius
M2: Specify drawing environment
M21: Identify the drawing type
M211: Identify line
M212: Identify rectangle
M213: Identify circle
M22: Detect drawing location
M221: Detect mouse push
M222: Detect mouse release
M23: Draw the element
S
C

FIG. 36

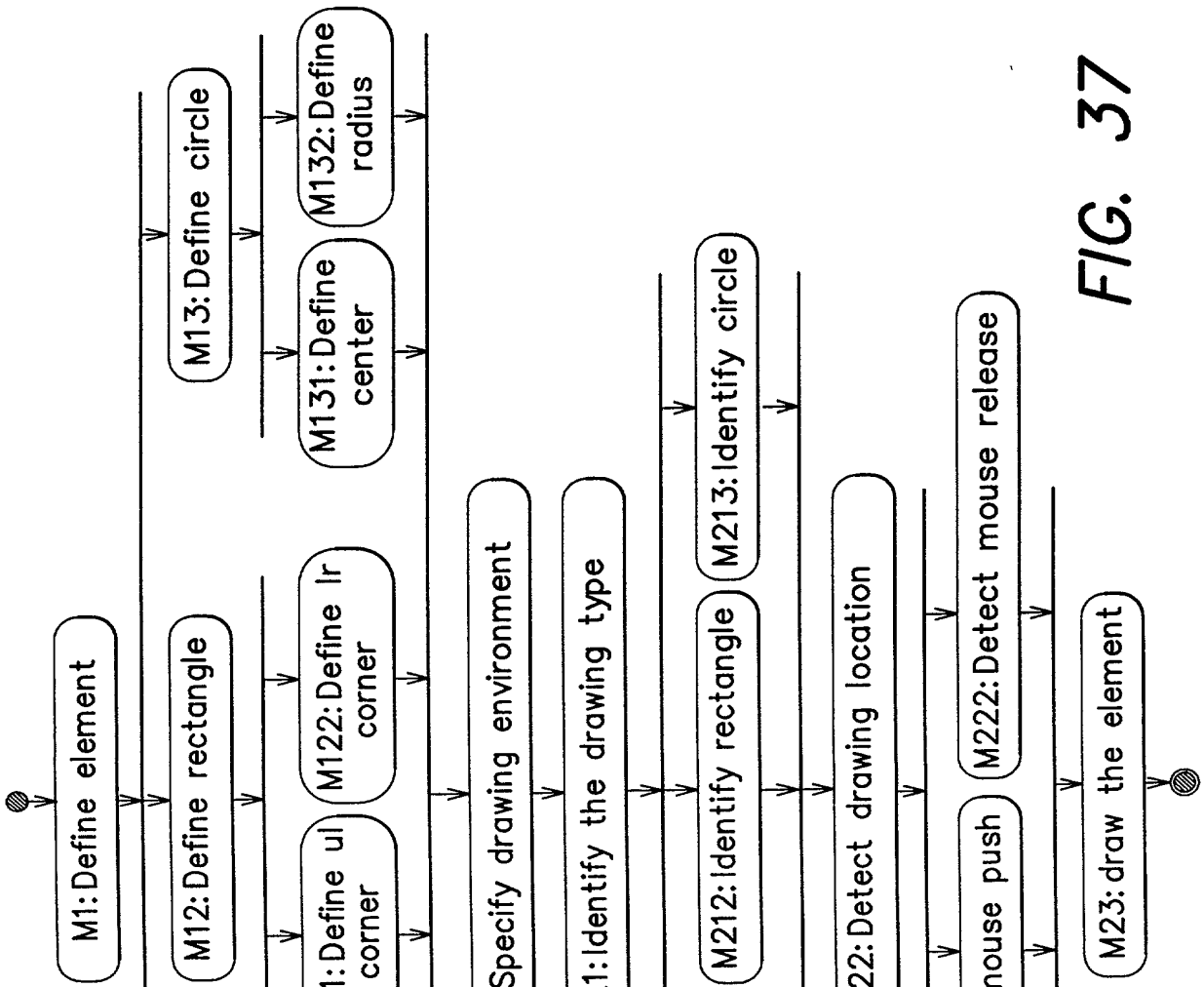

M1: Define element
M11: Define line
M111: Define start
M112: Define end
M12: Define rectangle
M121: Define ul corner
M122: Define lr corner
M13: Define circle
M131: Define center
M132: Define radius
M2: Specify drawing environment
M21: Identify the drawing type
M211: Identify line
M212: Identify rectangle
M213: Identify circle
M22: Detect drawing location
M221: Detect mouse push
M222: Detect mouse release
M23: draw the element

FIG. 37

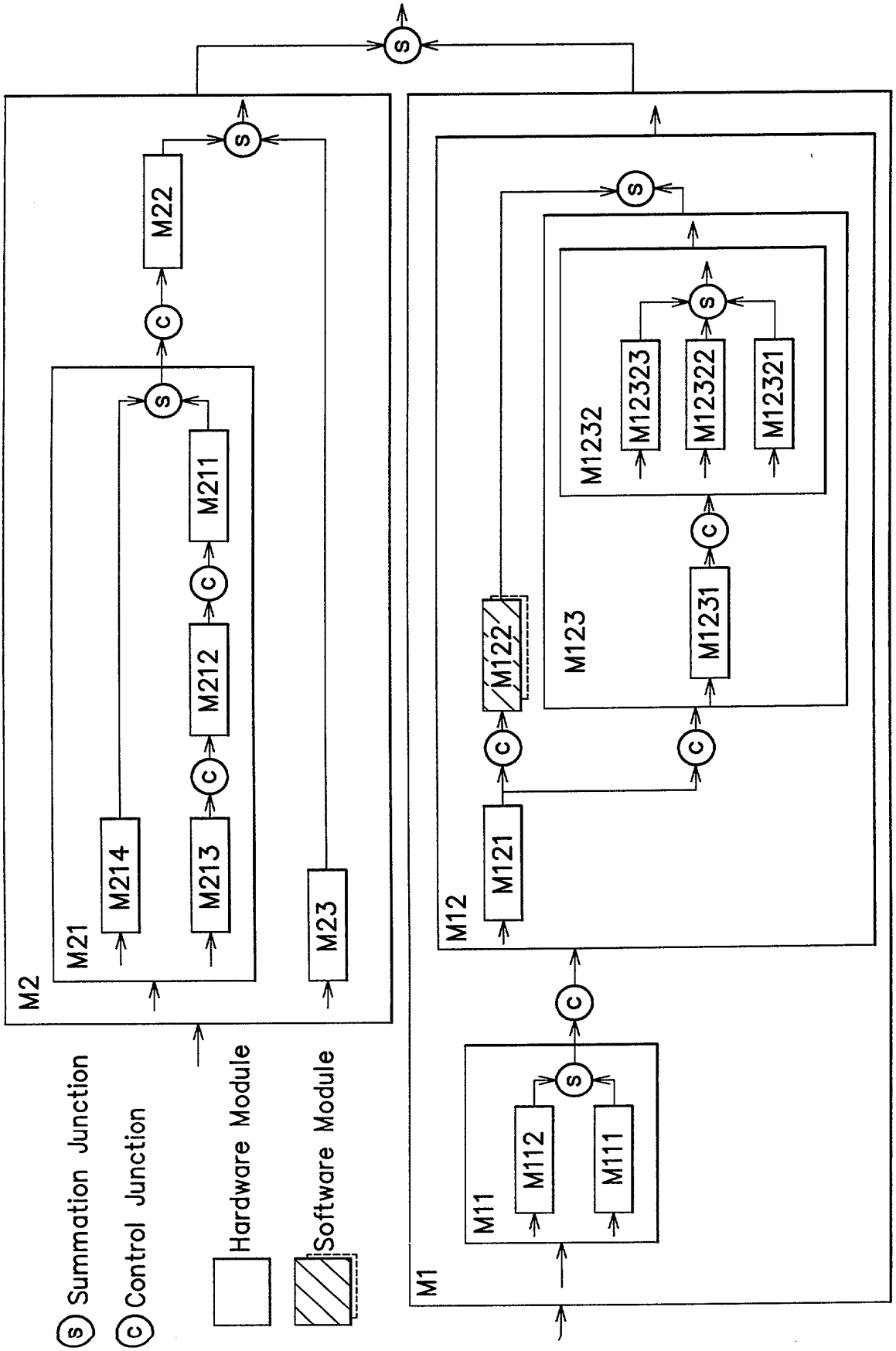

S Summation Junction
C Control Junction
Hardware Module
Software Module
M1
M11
M112
M111
M12
M121
M122
M123
M1231
M1232
M12323
M12322
M12321
M2
M21
M214
M213
M212
M211
M22
M23

FIG. 38

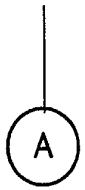
```
(object Petal
   version 40)
(object Design "Logical View"
   is_unit          TRUE
   is_loaded        TRUE
   file_name        "SDATA\\demo1.mdl"
   quid             "3353F13A0384"
   defaults         (object defaults
                    rightMargin          0.250000
                    leftMargin           0.250000
                    topMargin            0.250000
                    bottomMargin         0.500000
                    pageOverlap          0.250000
                    clipIconLabels       TRUE
                    autoResize           TRUE
                    snapToGrid           TRUE
                    gridX                16
                    gridY                16
                    defaultFont          (object Font
                       size              9
                       face              "helvetica"
                       bold              FALSE
                       italics           FALSE
                       underline         FALSE
                       strike            FALSE
                       color             0
                       default_color     TRUE)
```

FIG. 39A
FIG. 39B

A

B

FIG. 39A




```
                showMessageNum      1
                showClassOfObject   TRUE
                notation            "Unified")
root_usecase_package                (object Class_category"UseCaseView"
                quid                "3353F13A0386"
                exportControl       "Public"
                global              TRUE
                logical_models      (list unit_reference_list
                  (objectClass"Student"
                                    quid            "3353F162000A"
                                    documentation   "Someone who is registered to take classes at the University"
                                    stereotype      "Actor")
```

FIG. 39B

Code	Parent	Number	Description	Keyword	Comment	Category	Verification	Leaf
EX–a	0	1	Define element	—	—	—	—	FALSE
EX–a	0	2	Specify drawing environment	—	—	—	—	FALSE
EX–a	1	1	Define line element	—	—	—	—	FALSE
EX–a	1	2	Define rectangle element	—	—	—	—	FALSE
EX–a	1	3	Define circle element	—	—	—	—	FALSE
EX–a	1.1	1	Define start	—	—	—	—	TRUE
EX–a	1.1	2	Define end	—	—	—	—	TRUE
EX–a	1.2	1	Define upper left corner	—	—	—	—	TRUE
EX–a	1.2	2	Define lower right corner	—	—	—	—	TRUE
EX–a	1.3	1	Define center	—	—	—	—	TRUE
EX–a	1.3	2	Define radius	—	—	—	—	TRUE
EX–a	2	1	Identify the drawing type	—	—	—	—	FALSE
EX–a	2	2	Detect drawing location	—	—	—	—	FALSE
EX–a	2	3	Draw the element	—	—	—	—	TRUE
EX–a	2.1	1	Identify line	—	—	—	—	TRUE
EX–a	2.1	2	Identify rectangle	—	—	—	—	TRUE
EX–a	2.1	3	Identify circle	—	—	—	—	TRUE
EX–a	2.2	1	Detect mouse push	—	—	—	—	TRUE
EX–a	2.2	2	Detect mouse release	—	—	—	—	TRUE

FIG. 40

Code	Parent	Number	Alternative	Description	Keyword	Comment	Category	Verification	Leaf
EX–a	0	1	0	Element characteristics	—	—	—	—	FALSE
EX–a	0	2	0	GUI with window	—	—	—	—	FALSE
EX–a	1	1	0	Line characteristics	—	—	—	—	FALSE
EX–a	1	2	0	Rectangle characteristics	—	—	—	—	FALSE
EX–a	1	3	0	Circle characteristics	—	—	—	—	FALSE
EX–a	1.1	1	0	Start point	—	—	—	—	TRUE
EX–a	1.1	2	0	End point	—	—	—	—	TRUE
EX–a	1.2	1	0	Upper left point	—	—	—	—	TRUE
EX–a	1.2	2	0	Lower right point	—	—	—	—	TRUE
EX–a	1.3	1	0	Center point	—	—	—	—	TRUE
EX–a	1.3	2	0	Radius	—	—	—	—	TRUE
EX–a	2	1	0	Radio buttons	—	—	—	—	FALSE
EX–a	2	2	0	Mouse click information	—	—	—	—	FALSE
EX–a	2	3	0	Drawing area	—	—	—	—	TRUE
EX–a	2.1	1	0	Line button	—	—	—	—	TRUE
EX–a	2.1	2	0	Rectangle button	—	—	—	—	TRUE
EX–a	2.1	3	0	Circle button	—	—	—	—	TRUE
EX–a	2.2	1	0	Event for push	—	—	—	—	TRUE
EX–a	2.2	2	0	Event for release	—	—	—	—	TRUE

DP Table

(A) (B) (C) (D) (E) (F) (G) (H) (I) (J) (K) (L) (M) (N) (O) (P)

54/127

FIG. 41A	FIG. 41B

FIG. 41A

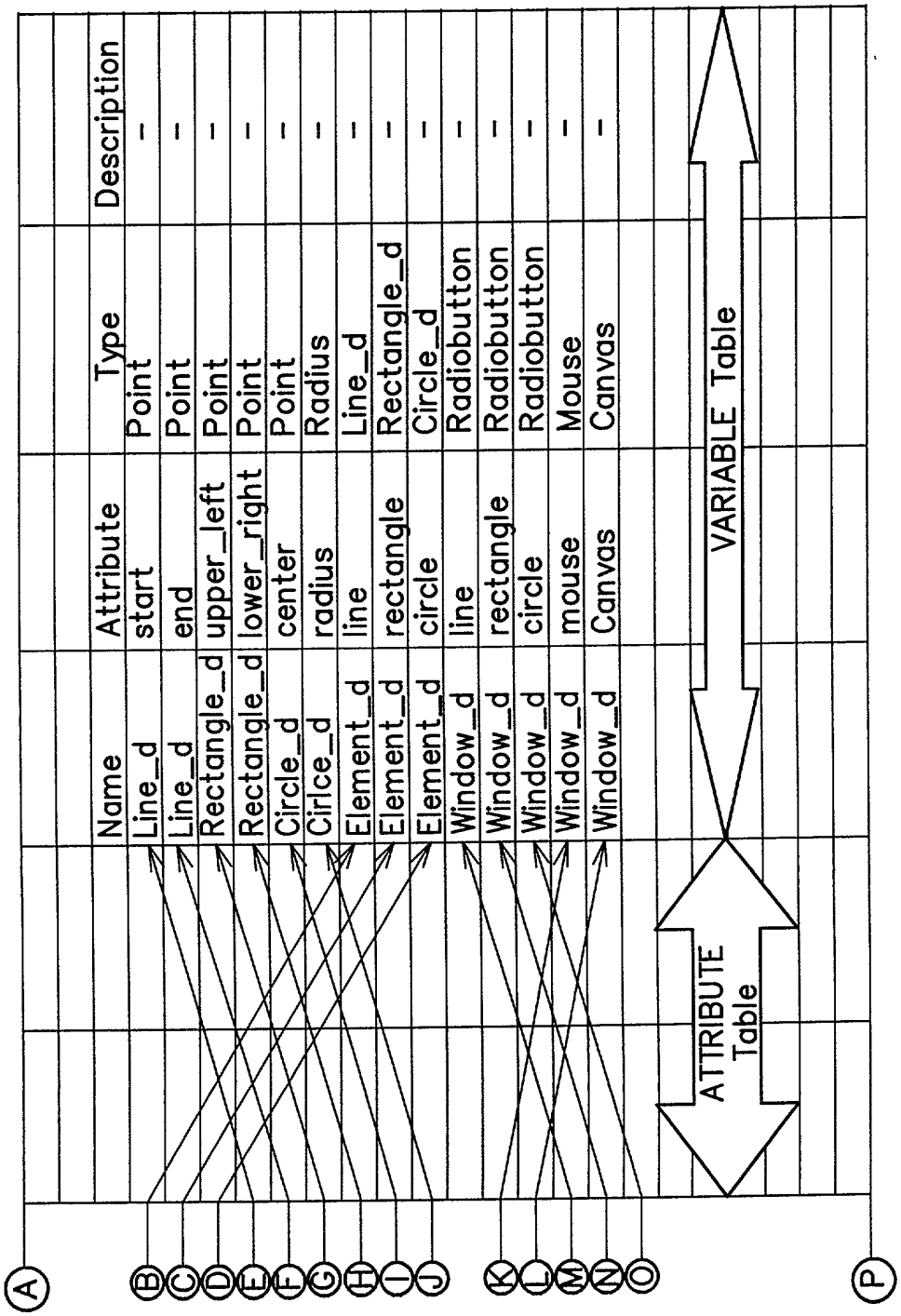

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Name
Attribute
Type
Description
Line_d
start
Point
--
Line_d
end
Point
--
Rectangle_d
upper_left
Point
--
Rectangle_d
lower_right
Point
--
Circle_d
center
Point
--
Cirlce_d
radius
Radius
--
Element_d
line
Line_d
--
Element_d
rectangle
Rectangle_d
--
Element_d
circle
Circle_d
--
Window_d
line
Radiobutton
--
Window_d
rectangle
Radiobutton
--
Window_d
circle
Radiobutton
--
Window_d
mouse
Mouse
--
Window_d
Canvas
Canvas
--
ATTRIBUTE Table
VARIABLE Table

FIG. 41B

Code1	Code2	Value	Comment
Ex–a.0.1	Ex–a.0.1.0	A	–
Ex–a.0.2	Ex–a.0.1.0	a	–
Ex–a.0.2	Ex–a.0.2.0	B	–
Ex–a.1.1	Ex–a.1.1.0	C	–
Ex–a.1.2	Ex–a.1.2.0	D	–
Ex–a.1.3	Ex–a.1.3.0	E	–
Ex–a.2.1	Ex–a.2.1.0	F	–
Ex–a.2.2	Ex–a.2.1.0	b	–
Ex–a.2.2	Ex–a.2.2.0	G	–
Ex–a.2.3	Ex–a.2.1.0	c	–
Ex–a.2.3	Ex–a.2.3.0	H	–
Ex–a.1.1.1	Ex–a.1.1.1.0	I	–
Ex–a.1.1.2	Ex–a.1.1.2.0	J	–
Ex–a.1.2.1	Ex–a.1.2.1.0	K	–
Ex–a.1.2.2	Ex–a.1.2.2.0	L	–
Ex–a.1.3.1	Ex–a.1.3.1.0	M	–
Ex–a.1.3.2	Ex–a.1.3.2.0	N	–
Ex–a.2.1.1	Ex–a.2.1.1.0	O	–
Ex–a.2.1.2	Ex–a.2.1.2.0	P	–
Ex–a.2.1.3	Ex–a.2.1.3.0	Q	–
Ex–a.2.2.1	Ex–a.2.2.1.0	R	–
Ex–a.2.2.2	Ex–a.2.2.2.0	S	–

Name	Method	Type	Description
LIne_d	LIne_d()	LIne_d	–
Line_d	setStart()	void	–
Line_d	setEnd()	void	–
Rectangle_d	Rectangle_d()	Rectangle_d	–
Rectangle_d	setULCorner()	void	–
Rectangle_d	setLRCorner()	void	–
Circle_d	Circle_d()	Circle_d	–
Circle_d	setCenter()	void	–
Circle_d	setRadius()	void	–
Element_d	Element_d()	Element_d	–
Window_d	Window_d()	Window_d	–
Window_d	CreateButtons()	void	–
Window_d	addLine()	void	–
Window_d	addRectangle()	void	–
Window_d	addCircle()	void	–
Window_d	MouseListener()	void	–
Window_d	mousePressed()	Point	–
Window_d	mouseReleased()	Point	–
Window_d	draw()	void	–
Window_d	isLineSelected()	boolean	–
Window_d	isRectangleSelected()	boolean	–
Window_d	isCircleSelected()	boolean	–

FIG. 42A
FIG. 42B

Ⓐ Ⓑ Ⓒ

FIG. 42A

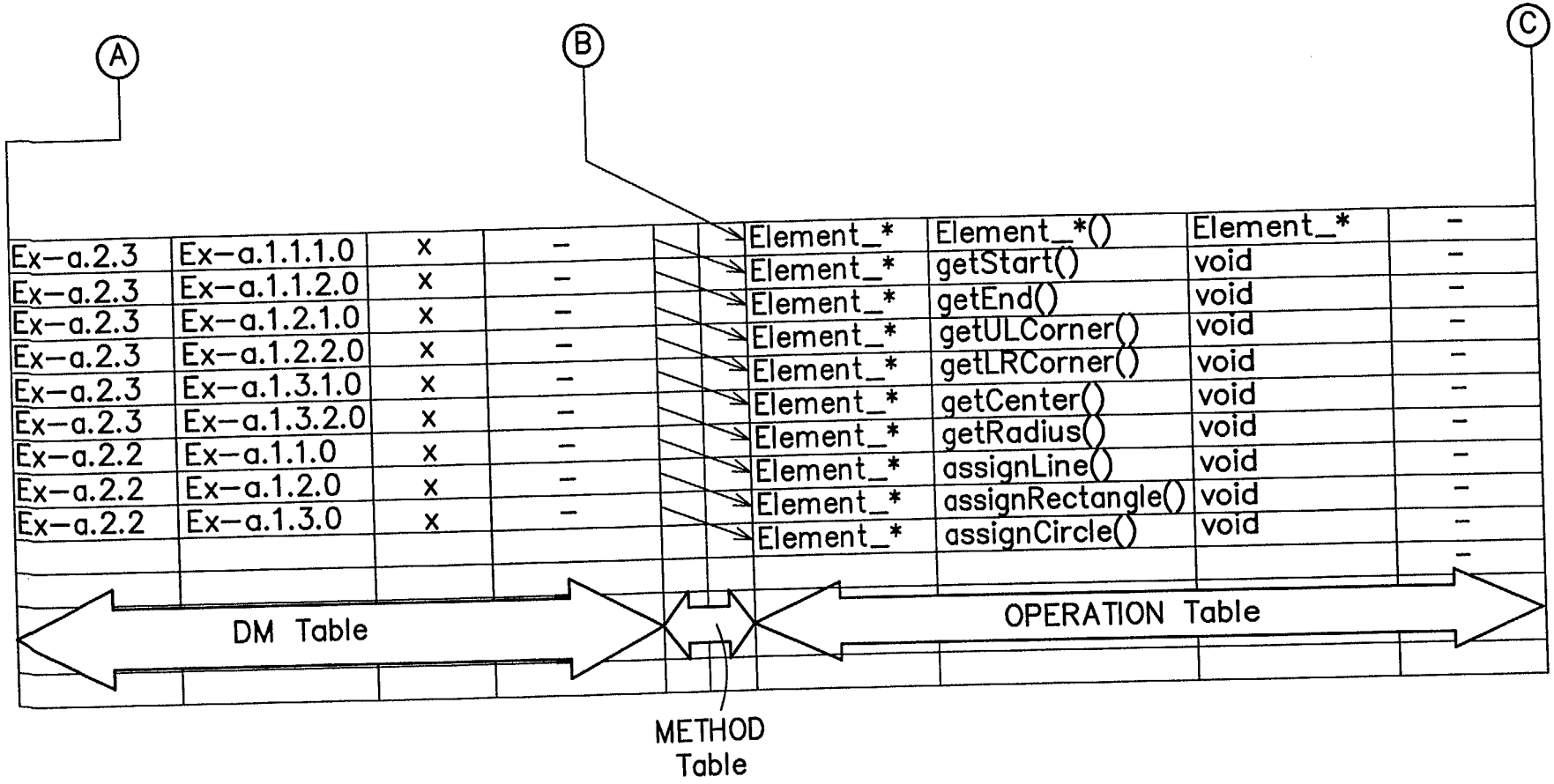

A
B
C
Ex–a.2.3 Ex–a.1.1.1.0 x –
Ex–a.2.3 Ex–a.1.1.2.0 x –
Ex–a.2.3 Ex–a.1.2.1.0 x –
Ex–a.2.3 Ex–a.1.2.2.0 x –
Ex–a.2.3 Ex–a.1.3.1.0 x –
Ex–a.2.3 Ex–a.1.3.2.0 x –
Ex–a.2.2 Ex–a.1.1.0 x –
Ex–a.2.2 Ex–a.1.2.0 x –
Ex–a.2.2 Ex–a.1.3.0 x –
Element_* Element_*() Element_* –
Element_* getStart() void –
Element_* getEnd() void –
Element_* getULCorner() void –
Element_* getLRCorner() void –
Element_* getCenter() void –
Element_* getRadius() void –
Element_* assignLine() void –
Element_* assignRectangle() void –
Element_* assignCircle() void –
DM Table
OPERATION Table
METHOD Table

FIG. 42B

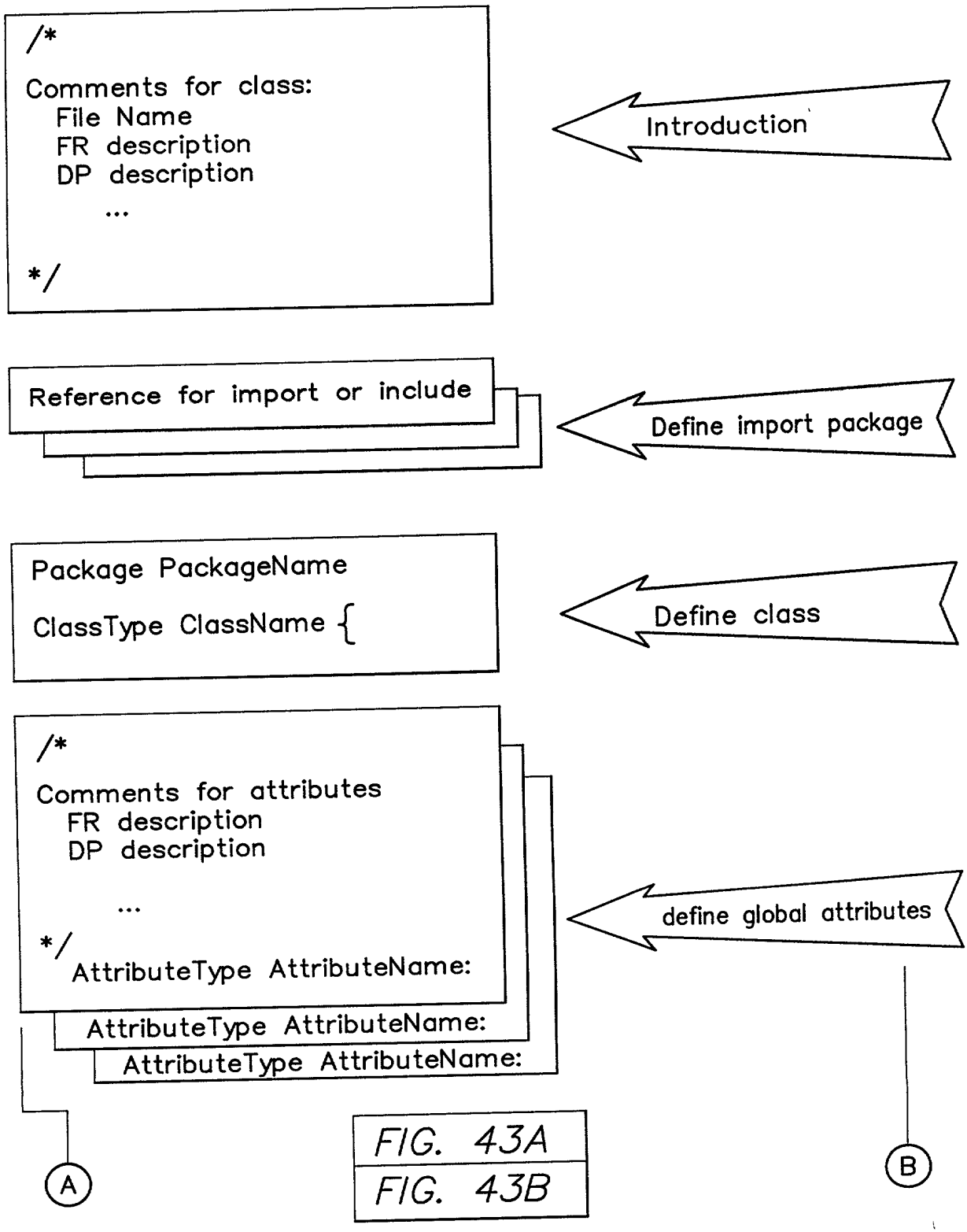


FIG. 43A
FIG. 43B

FIG. 43A

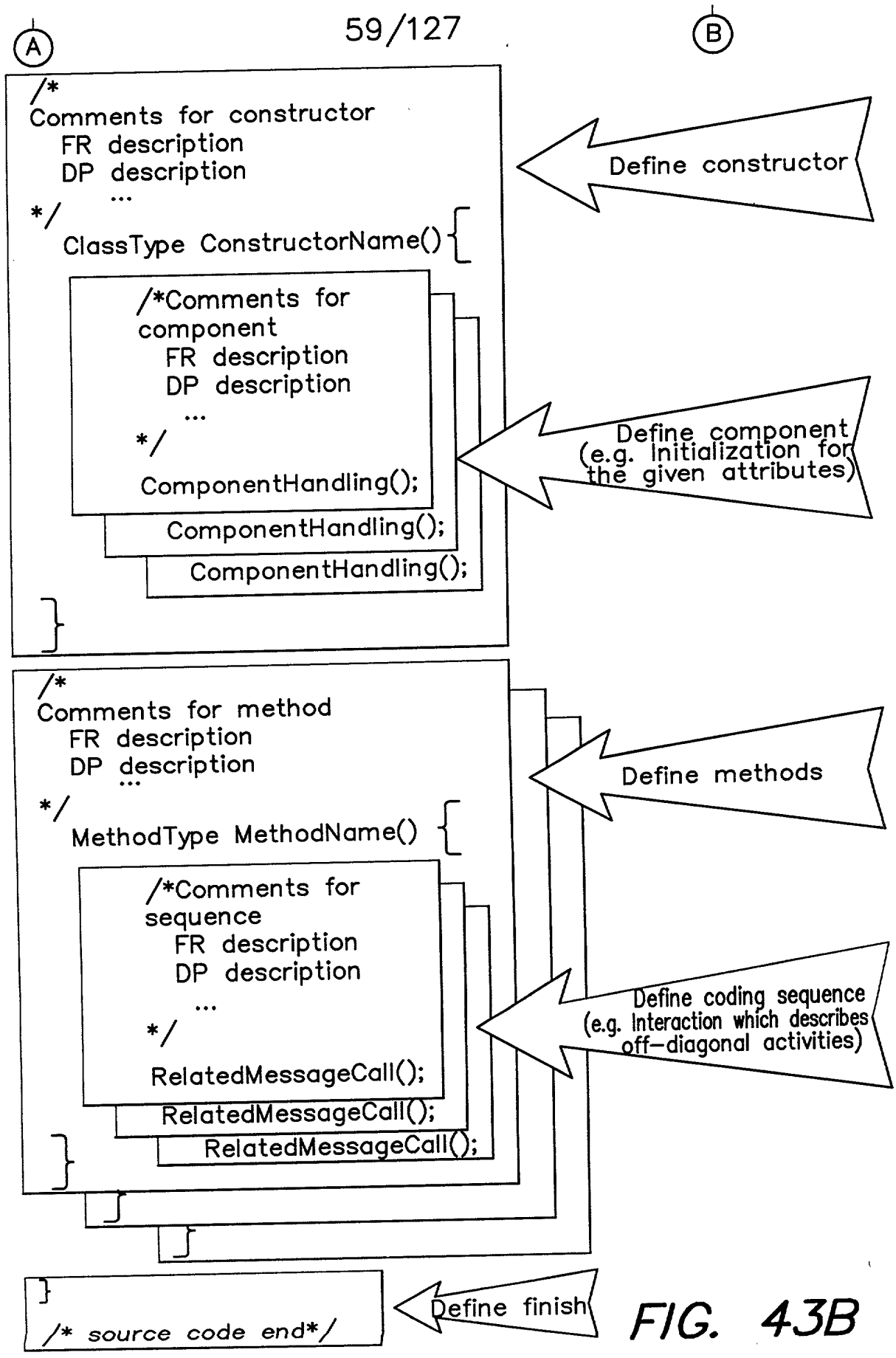

A
B
/*
Comments for constructor
FR description
DP description
...
*/
ClassType ConstructorName(){
/*Comments for component
FR description
DP description
...
*/
ComponentHandling();
ComponentHandling();
ComponentHandling();
}
Define constructor
Define component (e.g. Initialization for the given attributes)
/*
Comments for method
FR description
DP description
...
*/
MethodType MethodName() {
/*Comments for sequence
FR description
DP description
...
*/
RelatedMessageCall();
RelatedMessageCall();
RelatedMessageCall();
}
Define methods
Define coding sequence (e.g. Interaction which describes off-diagonal activities)
}
/* source code end*/
Define finish

FIG. 43B

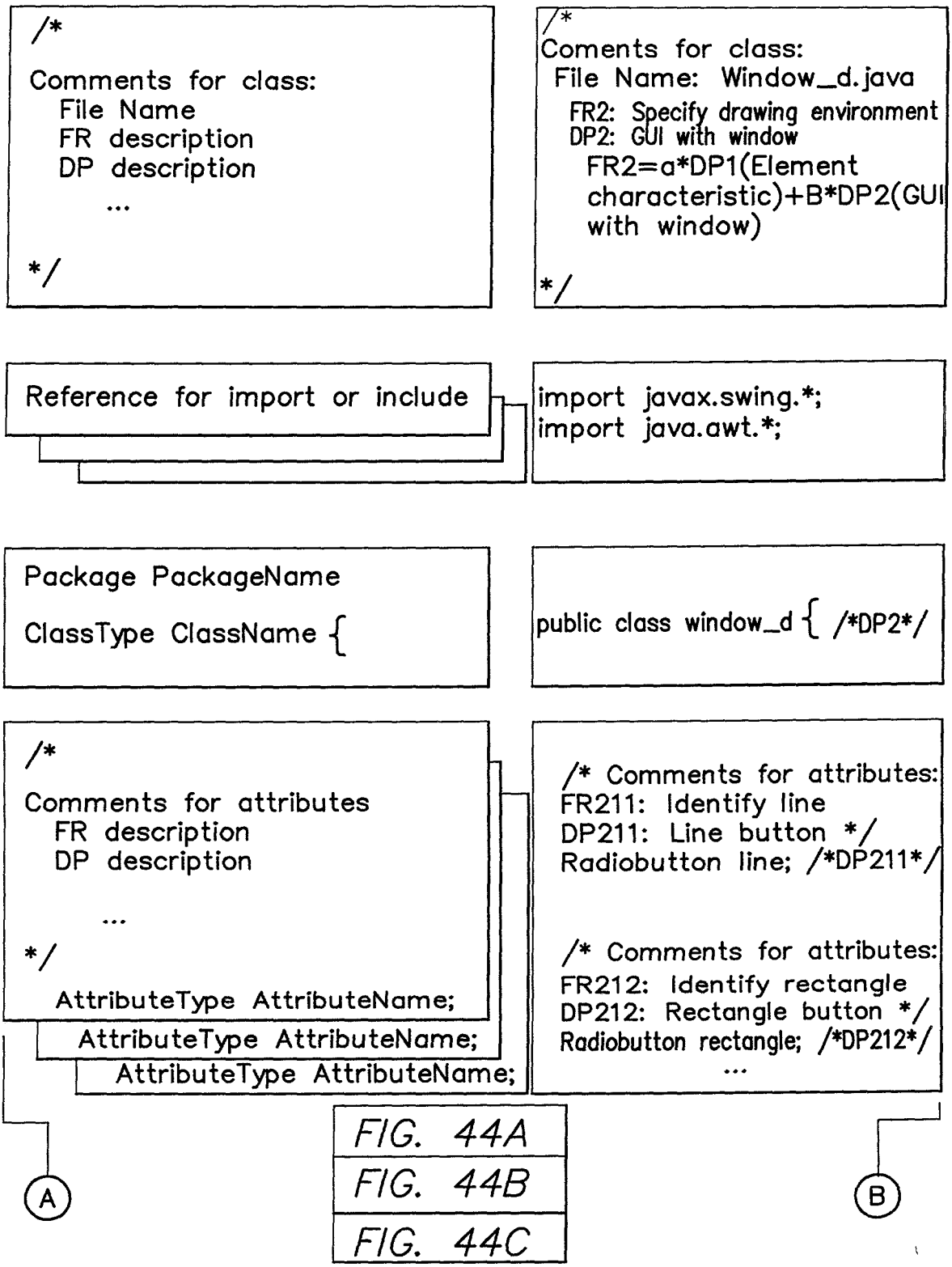


FIG. 44A
FIG. 44B
FIG. 44C

FIG. 44A

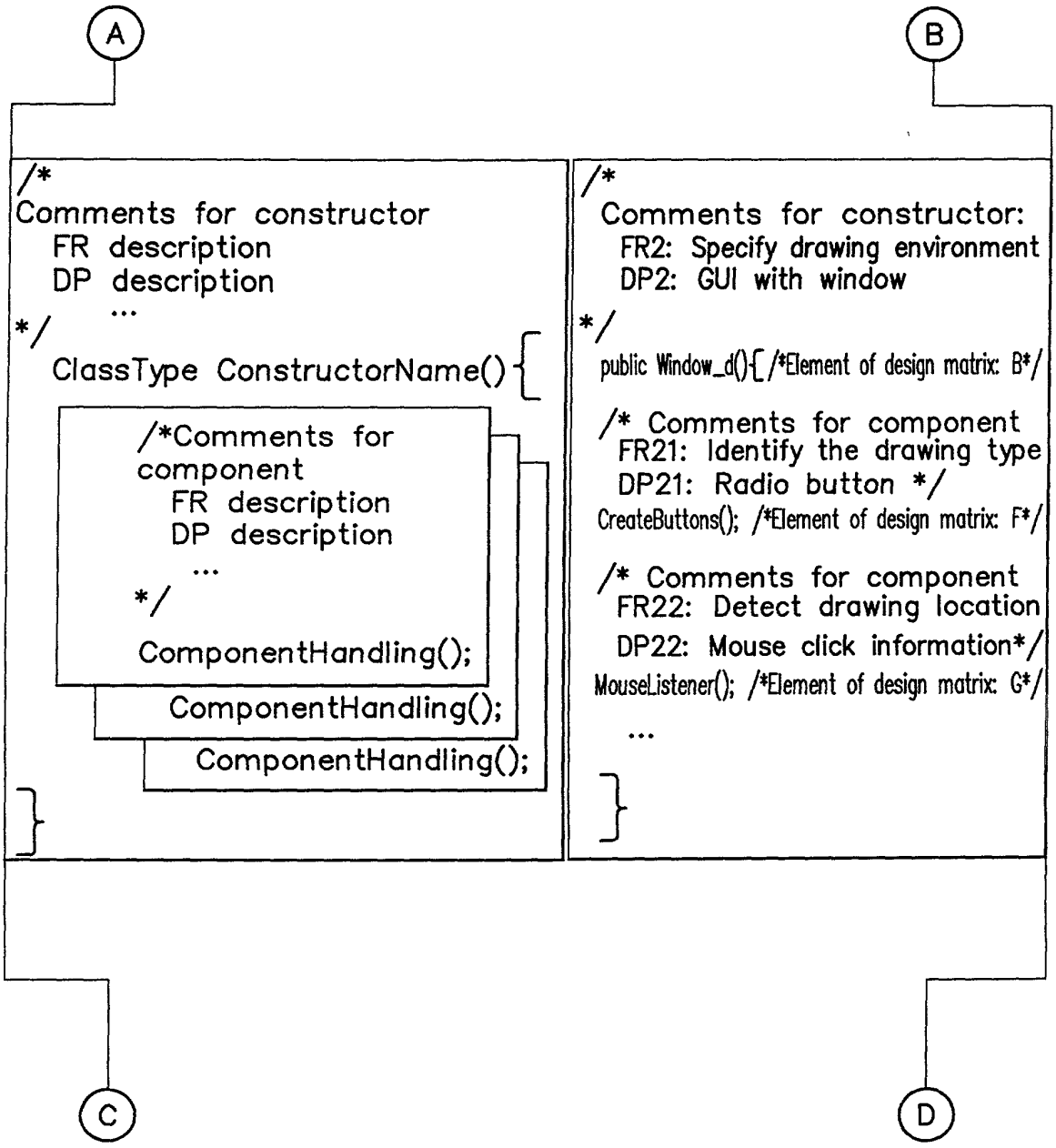

A
B
/*
Comments for constructor
FR description
DP description
...
*/
ClassType ConstructorName(){
/*Comments for component
FR description
DP description
...
*/
ComponentHandling();
ComponentHandling();
ComponentHandling();
}
/*
Comments for constructor:
FR2: Specify drawing environment
DP2: GUI with window
*/
public Window_d(){/*Element of design matrix: B*/
/* Comments for component
FR21: Identify the drawing type
DP21: Radio button */
CreateButtons(); /*Element of design matrix: F*/
/* Comments for component
FR22: Detect drawing location
DP22: Mouse click information*/
MouseListener(); /*Element of design matrix: G*/
...
}
C
D

FIG. 44B

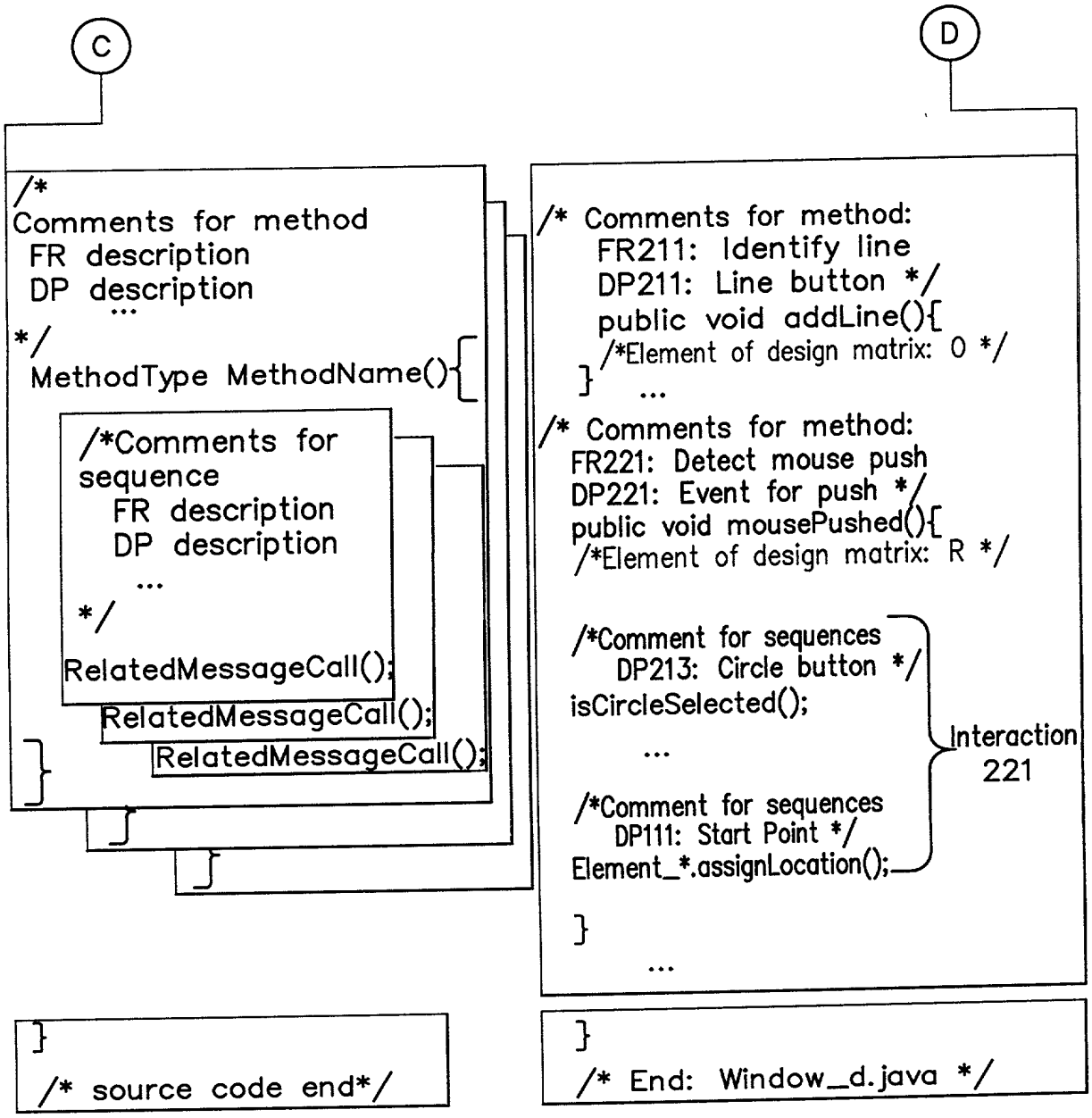

C
D
/*
Comments for method
FR description
DP description
...
*/
MethodType MethodName(){
/*Comments for sequence
FR description
DP description
...
*/
RelatedMessageCall();
RelatedMessageCall();
RelatedMessageCall();
}
/* source code end*/
/* Comments for method:
FR211: Identify line
DP211: Line button */
public void addLine(){
/*Element of design matrix: 0 */
}
...
/* Comments for method:
FR221: Detect mouse push
DP221: Event for push */
public void mousePushed(){
/*Element of design matrix: R */
/*Comment for sequences
DP213: Circle button */
isCircleSelected();
...
/*Comment for sequences
DP111: Start Point */
Element_*.assignLocation();
Interaction 221
}
...
}
/* End: Window_d.java */

FIG. 44C

FR Information:

Number	Description
FR#.1	Provide security
FR#.2	Assign tasks
FR#.3	Manage schedule
FR#.4	Construct design h...
FR#.5	Facilitate changes...

DP Information:

Num...	Description
DP#.1	Login privilege
DP#.2	Resource of desig...
DP#.3	Schedule—manage...
DP#.4	Data structure for...
DP#.5	ECO handling tool

FIG. 45A

	FR	DP
1	FR 1 description	<→ DP 1 description
2	FR 2 description	<→ DP 2 description
3	FR 3 description	<→ DP 3 description

FIG. 45B

FR Information:

Number	Description
FR#.1	Control the water fl...
FR#.2	Control the temper...

DP Information:

Number	Description
DP#.1	Angle for flow ra...
DP#.1(1)	Angle of hot wat...
DP#.2	Angle for tempe...
DP#.2(1)	Connecting rod...
DP#.2(2)	Angle of cold w...

FIG. 46A

	FR	DP
1	FR 1 description	DP 1 description
2	FR 2 description	Alternative DP 2(a)
		Alternative DP 2(b)
		Alternative DP 2(c)
3	FR 3 description	DP 3 description

FIG. 46B

Parent Information:

Number	Description
FR 1.1	Manage design workflow
DP 1.1	Management roadmap

FR Information:

Number	Description
FR#.1	Provide security
FR#.2	Assign tasks
FR#.3	Manage schedule
FR#.4	Construct design h...
FR#.5	Facilitate changes...

DP Information:

Number	Description
DP#.1	Login privilege
DP#.2	Resource of de...
DP#.3	Schedule-mana...
DP#.4	Data structure f...
DP#.5	ECO handling t...

FIG. 47A

	FR	DP
Parent	Parent FR description	Parent DP description
1	FR 1 description	DP 1 description
2	FR 2 description	Alternative DP 2(a)
		Alternative DP 2(b)
		Alternative DP 2(c)
3	FR 3 description	DP 3 description

FIG. 47B

Parent Information:

Number	Description
FR 1.1	Manage design workflow
DP 1.1	Management roadmap

FR Information:

Number	Description
FR#.1	Provide security
FR#.2	Assign tasks
FR#.3	Manage schedule
FR#.4	Construct design h...
FR#.5	Facilitate changes...

DP Information:

Number	Description
DP#.1	Login privilege
DP#.2	Resource of de...
DP#.3	Schedule—mana...
DP#.4	Data structure f...
DP#.5	ECO handling t...

FIG. 48A

#: 1.2.3	FR	DP
Parent	Parent FR description	Parent DP description
#.1	FR 1 description	DP 1 description
#.2	FR 2 description	Alternative DP 2(a)
		Alternative DP 2(b)
		Alternative DP 2(c)
#.3	FR 3 description	DP 3 description

FIG. 48B

Constraint Information:

Num...	Descr...	FR#.1	FR#.2	FR#.3	FR#.4	FR#.5
C#.1	Make...	X	X	X	X	X
C#.2	Supp...	X	X	X	X	X
C#.3	Elimi...	X	X	X	X	X
C#.4	Facilit...	X	X	X	X	X
C#.5	Funct...			X	X	
C#.6	Obie...			X	X	

FIG. 49A

Mapping | Constraints | Robust Design | Analysis

Index	Information				Target value		
#	Category	Type	Constraints	Comments	Operator	Target	Calculated
1	Critical	Marke	Weight	☑	Less than (<+)	300lb	
2	Interface	Field	Cost	☐	More than(>=)	$500	
3	Project	Manu	Volume	☑	Exact (=+/−)	10cu	

	C#.1	C#.2	C#.3	CA's
FR#.1	X			☑
FR#.2	X	X	X	☐
FR#.3	X		X	☐
FR#.4	X			☑

FIG. 49B

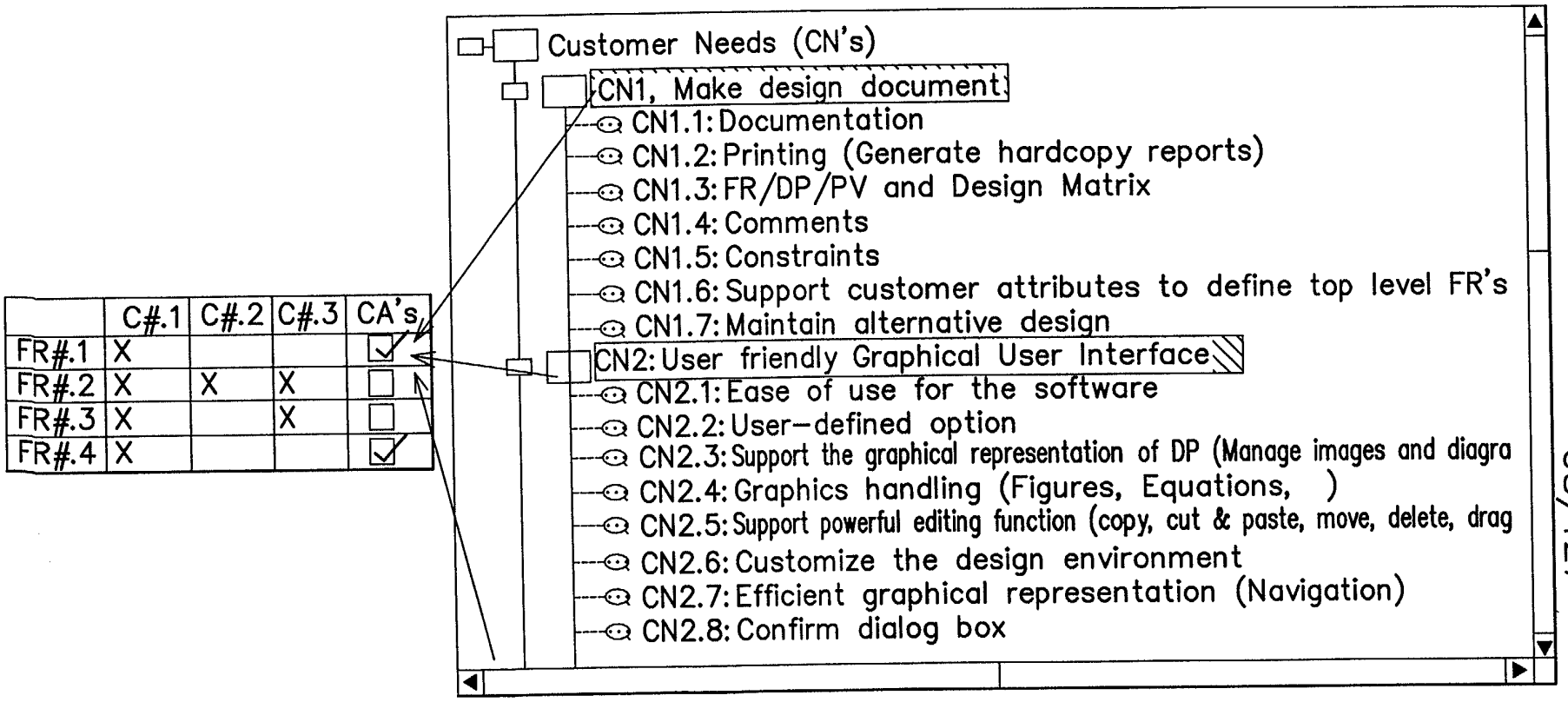

Customer Needs (CN's)
CN1, Make design document
CN1.1: Documentation
CN1.2: Printing (Generate hardcopy reports)
CN1.3: FR/DP/PV and Design Matrix
CN1.4: Comments
CN1.5: Constraints
CN1.6: Support customer attributes to define top level FR's
CN1.7: Maintain alternative design
CN2: User friendly Graphical User Interface
CN2.1: Ease of use for the software
CN2.2: User-defined option
CN2.3: Support the graphical representation of DP (Manage images and diagra
CN2.4: Graphics handling (Figures, Equations,)
CN2.5: Support powerful editing function (copy, cut & paste, move, delete, drag
CN2.6: Customize the design environment
CN2.7: Efficient graphical representation (Navigation)
CN2.8: Confirm dialog box
C#.1
C#.2
C#.3
CA's
FR#.1
FR#.2
FR#.3
FR#.4

FIG. 50

Index	Information				Target Value		
#	Category	Type	Constraints	Comments	Operator	Target	Calculated
1	Critical	Marke	Weight	☑	Less than (<+)	300lb	
2	Interface	Field	Cost	☐	More than(>=)	$500	
3	Project	Manu	Volume	☑	Exact (=+/−)	10cu	

FIG. 51

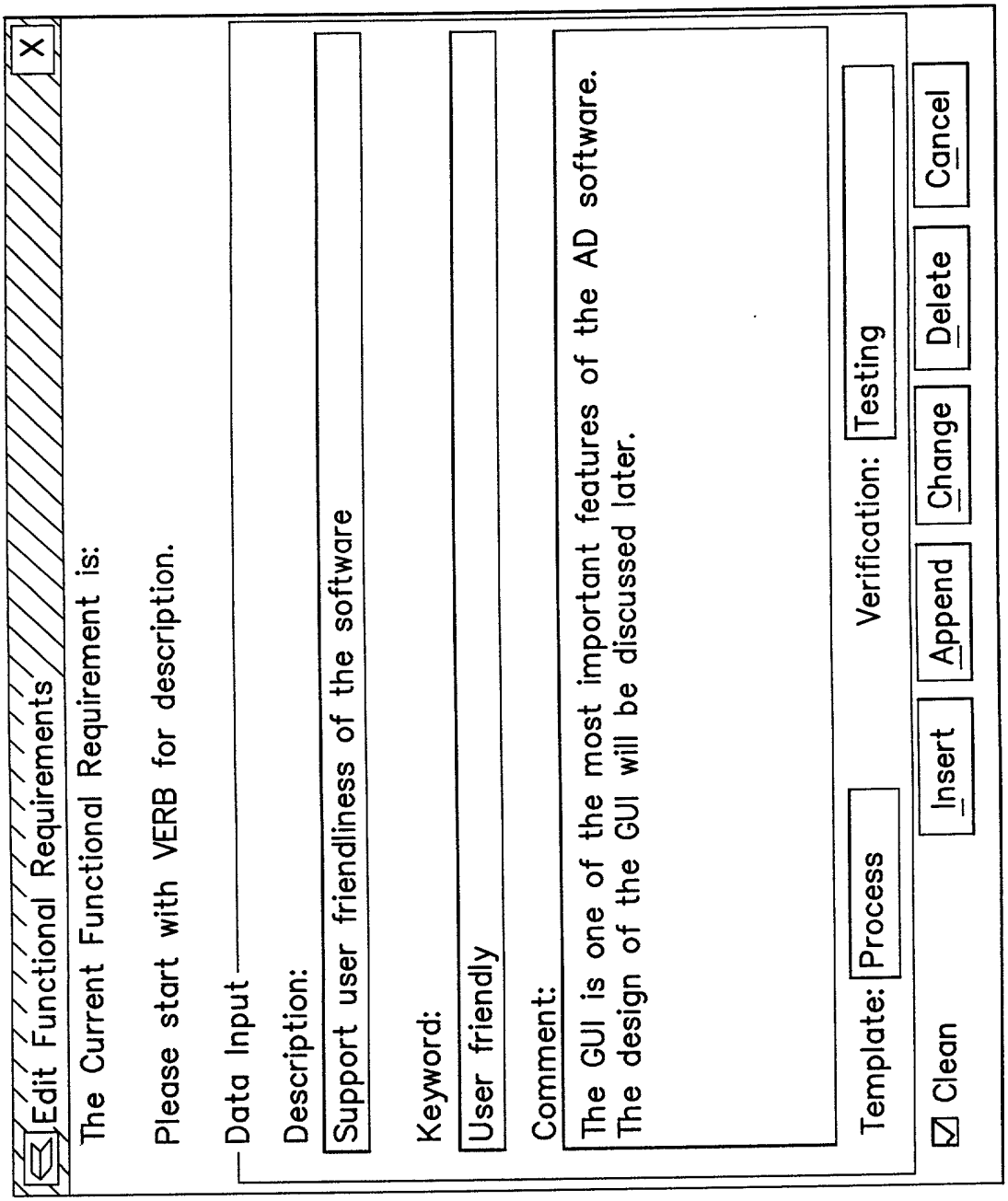

Edit Functional Requirements
The Current Functional Requirement is:
Please start with VERB for description.
Data Input
Description:
Support user friendliness of the software
Keyword:
User friendly
Comment:
The GUI is one of the most important features of the AD software.
The design of the GUI will be discussed later.
Template: Process
Verification: Testing
Clean
Insert
Append
Change
Delete
Cancel

FIG. 52A

Parent Information

Nu...	Description	Comment
FR 1	Make a decision-making tool whi...	A software tool for decision maki...
DP 1	Computerized system with the A...	Software for Axiomatic Design.

FR Information:

Num...	Description	Comment
FR#.1	Manage desi...	The design a...
FR#.2	Provide decis...	The FR deal...
FR#.3	Support user...	The GUI is a...
FR#.4	Provide effici...	All kinds of d...
FR#.5	Provide utility...	The fundam...

DP Information:

Num...	Description	Comm...
DP#.1	Management ro...	
DP#.2	Decision-maki...	
DP#.3	Graphical User...	
DP#.4	Data-managing...	
DP#.5	Plug-in software	

FIG. 52B

Index #	Template	Information		Comment		App. Link
		FR	DP	FR	DP	
Parent		Control the FR/DP domain	FR/DP window	☑	☐	
1		Control the mapping	Mapping tab	☑	☑	
			Domain tab	☐	☑	
2		Assign constraints	Constraints tab	☐	☐	
3		Refine the design	Robust design tab	☑	☐	
4		Analyze the design	Analysis tab	☑	☑	

FIG. 52C

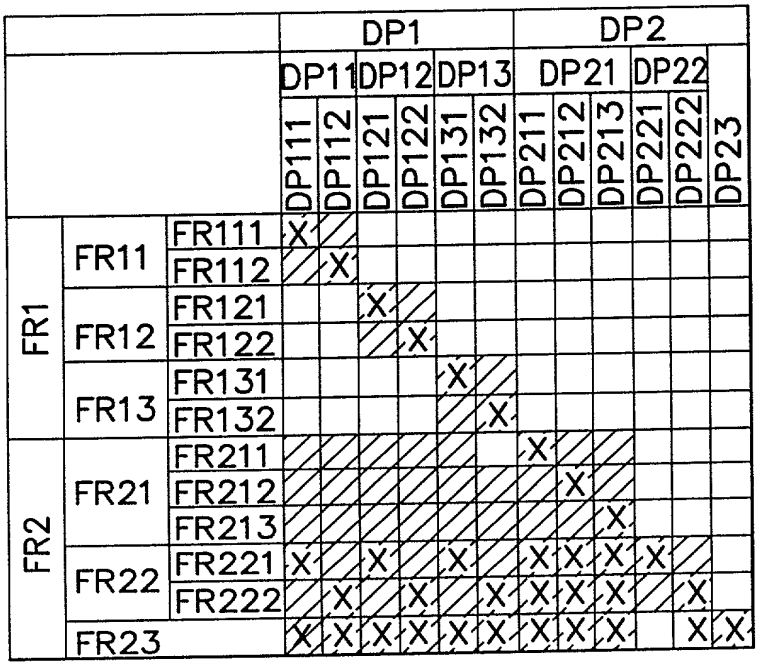

DP1
DP2
DP11
DP12
DP13
DP21
DP22
DP111
DP112
DP121
DP122
DP131
DP132
DP211
DP212
DP213
DP221
DP222
DP23
FR1
FR2
FR11
FR12
FR13
FR21
FR22
FR23
FR111
FR112
FR121
FR122
FR131
FR132
FR211
FR212
FR213
FR221
FR222

FIG. 53

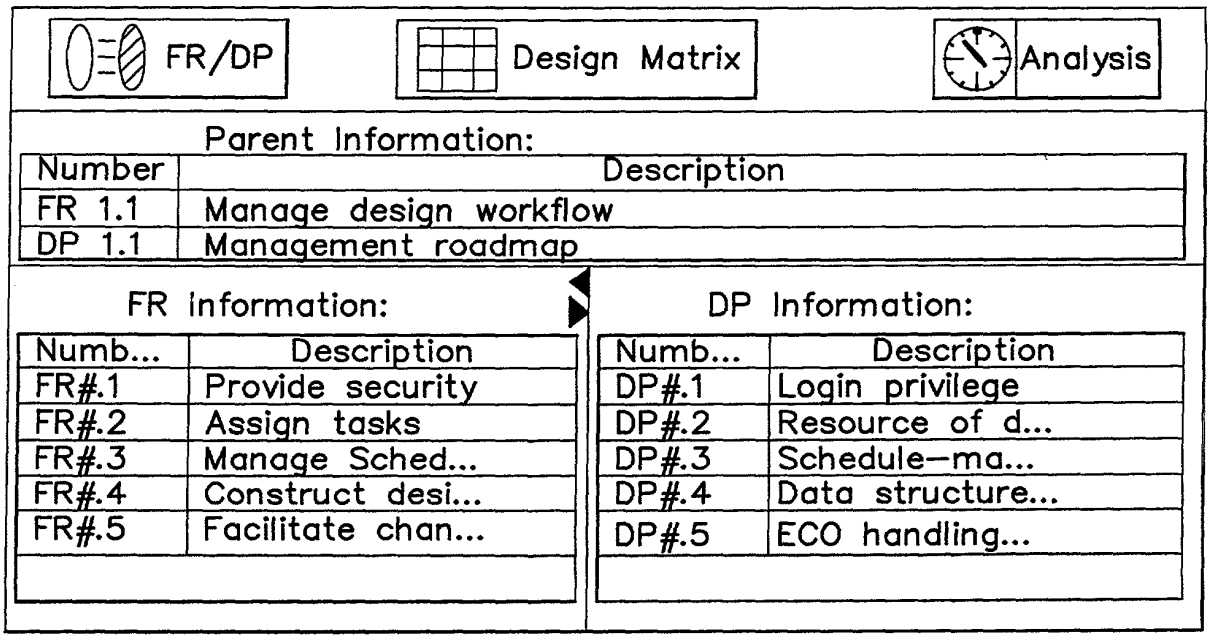


FIG. 54A

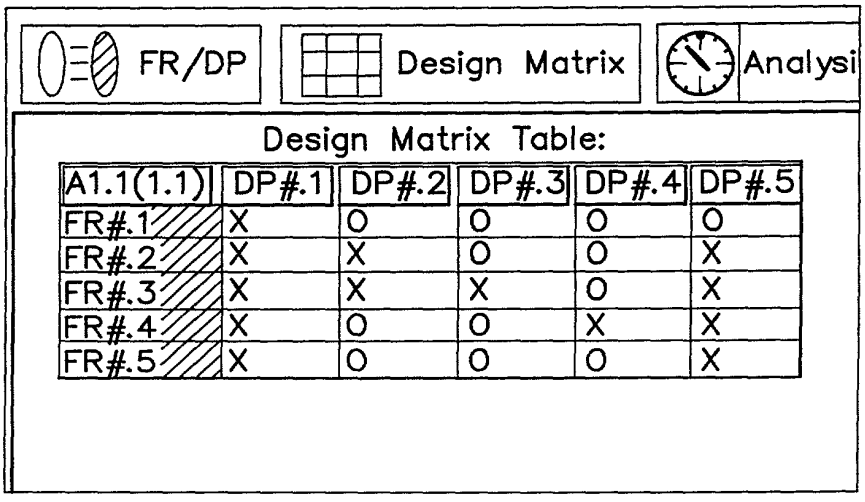
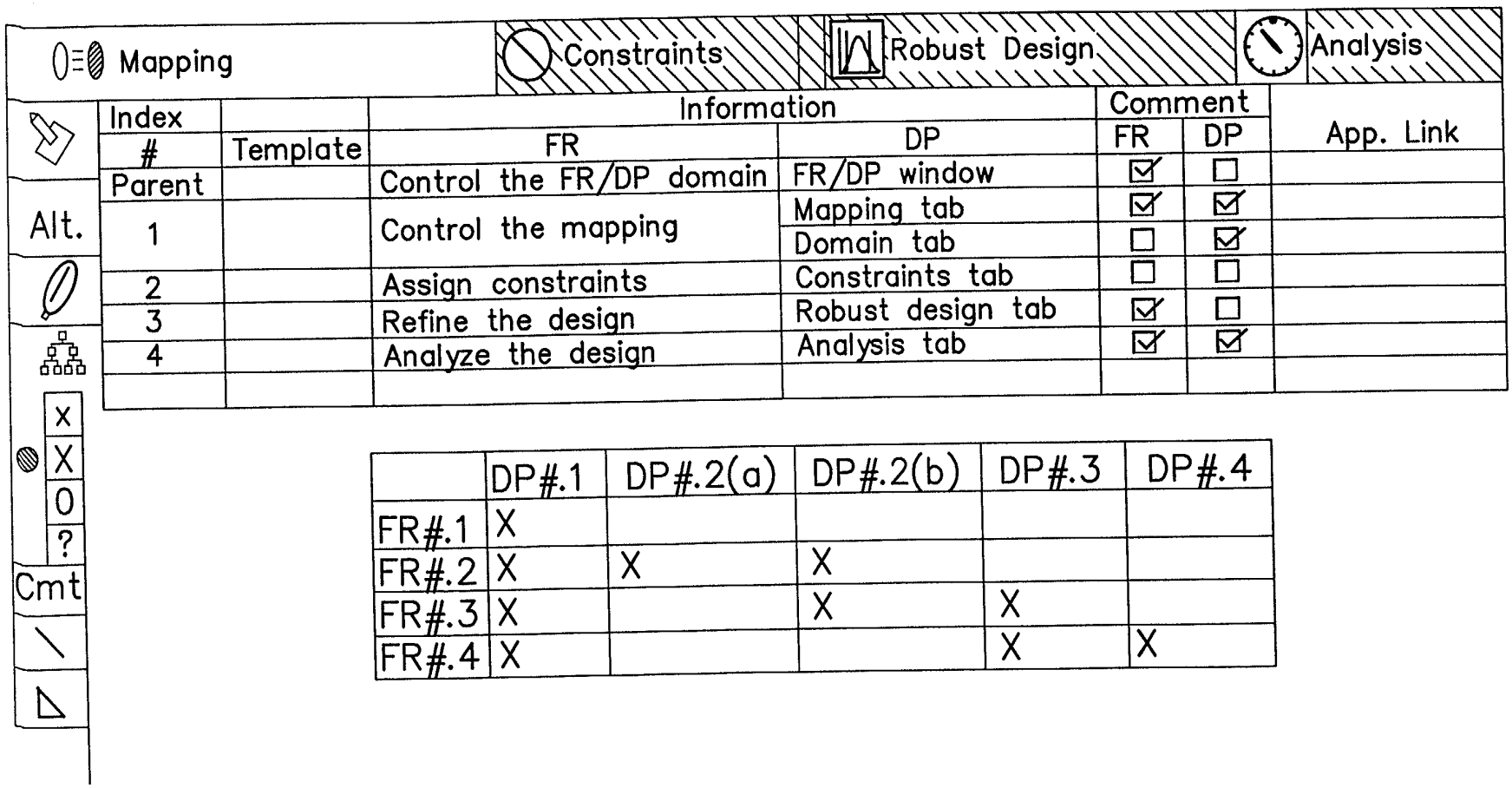


FIG. 54B



Index #	Template	Information FR	Information DP	Comment FR	Comment DP	App. Link
Parent		Control the FR/DP domain	FR/DP window	☑	☐	
1		Control the mapping	Mapping tab	☑	☑	
			Domain tab	☐	☑	
2		Assign constraints	Constraints tab	☐	☐	
3		Refine the design	Robust design tab	☑	☐	
4		Analyze the design	Analysis tab	☑	☑	

	DP#.1	DP#.2(a)	DP#.2(b)	DP#.3	DP#.4
FR#.1	X				
FR#.2	X	X	X		
FR#.3	X		X	X	
FR#.4	X			X	X

FIG. 54C

FIG. 55A

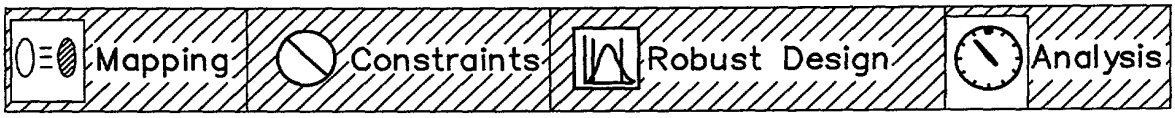


FIG. 55B

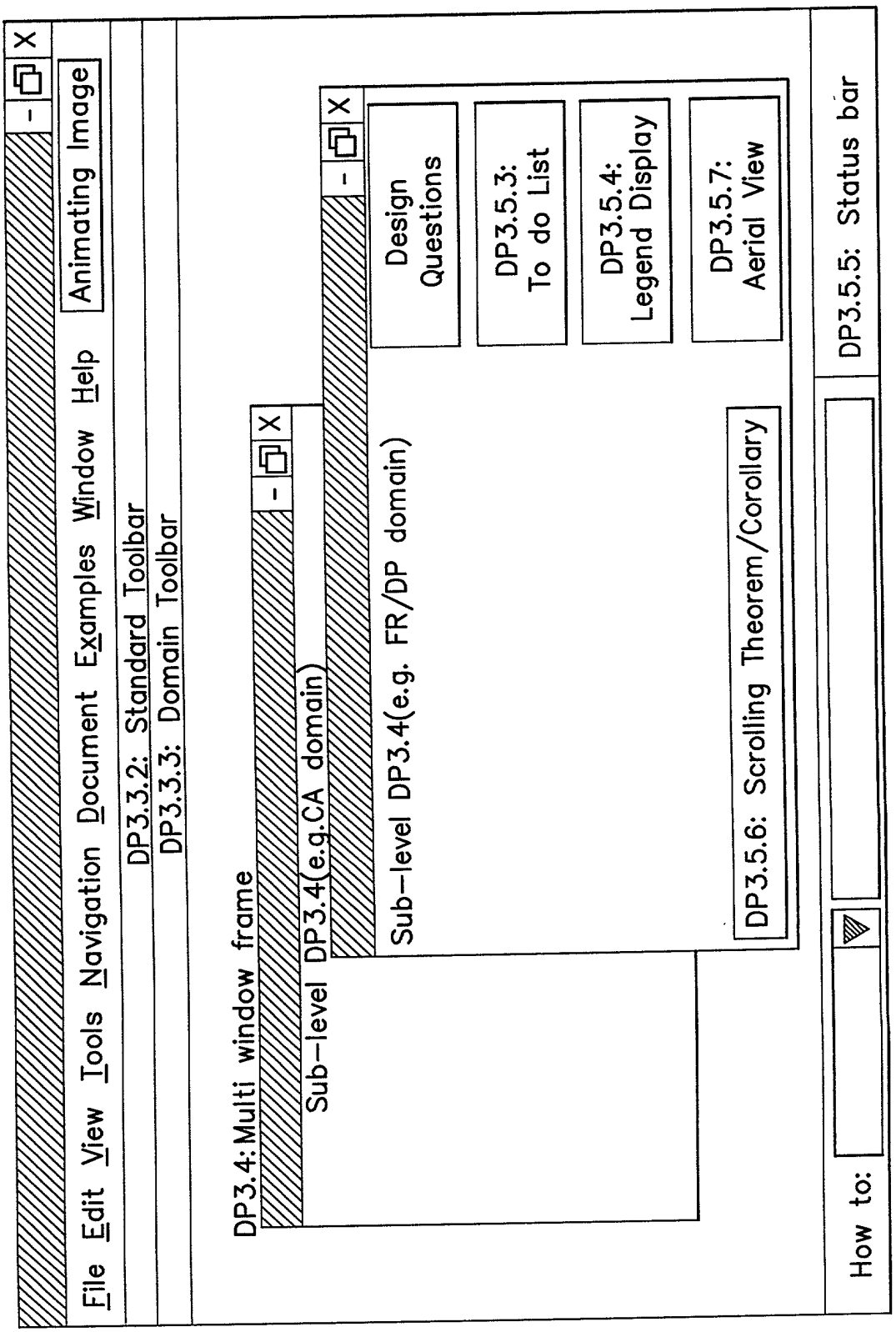

File Edit View Tools Navigation Document Examples Window Help
Animating Image
DP3.3.2: Standard Toolbar
DP3.3.3: Domain Toolbar
DP3.4: Multi window frame
Sub-level DP3.4(e.g.CA domain)
Sub-level DP3.4(e.g. FR/DP domain)
Design Questions
DP3.5.3: To do List
DP3.5.4: Legend Display
DP3.5.7: Aerial View
DP3.5.6: Scrolling Theorem/Corollary
How to:
DP3.5.5: Status bar

FIG. 56

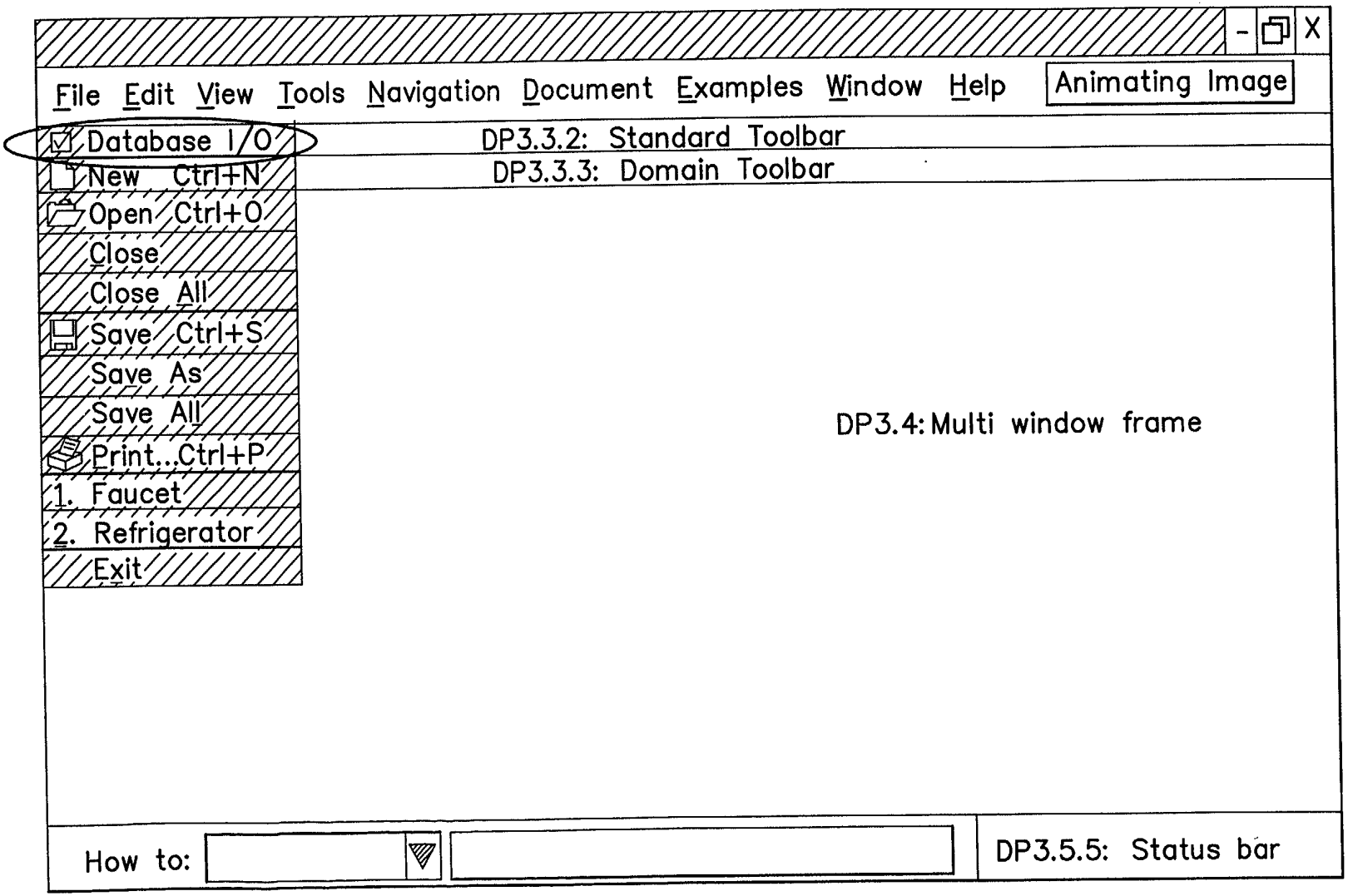

File Edit View Tools Navigation Document Examples Window Help
Animating Image
Database I/O
New Ctrl+N
Open Ctrl+O
Close
Close All
Save Ctrl+S
Save As
Save All
Print...Ctrl+P
1. Faucet
2. Refrigerator
Exit
DP3.3.2: Standard Toolbar
DP3.3.3: Domain Toolbar
DP3.4: Multi window frame
How to:
DP3.5.5: Status bar

FIG. 57

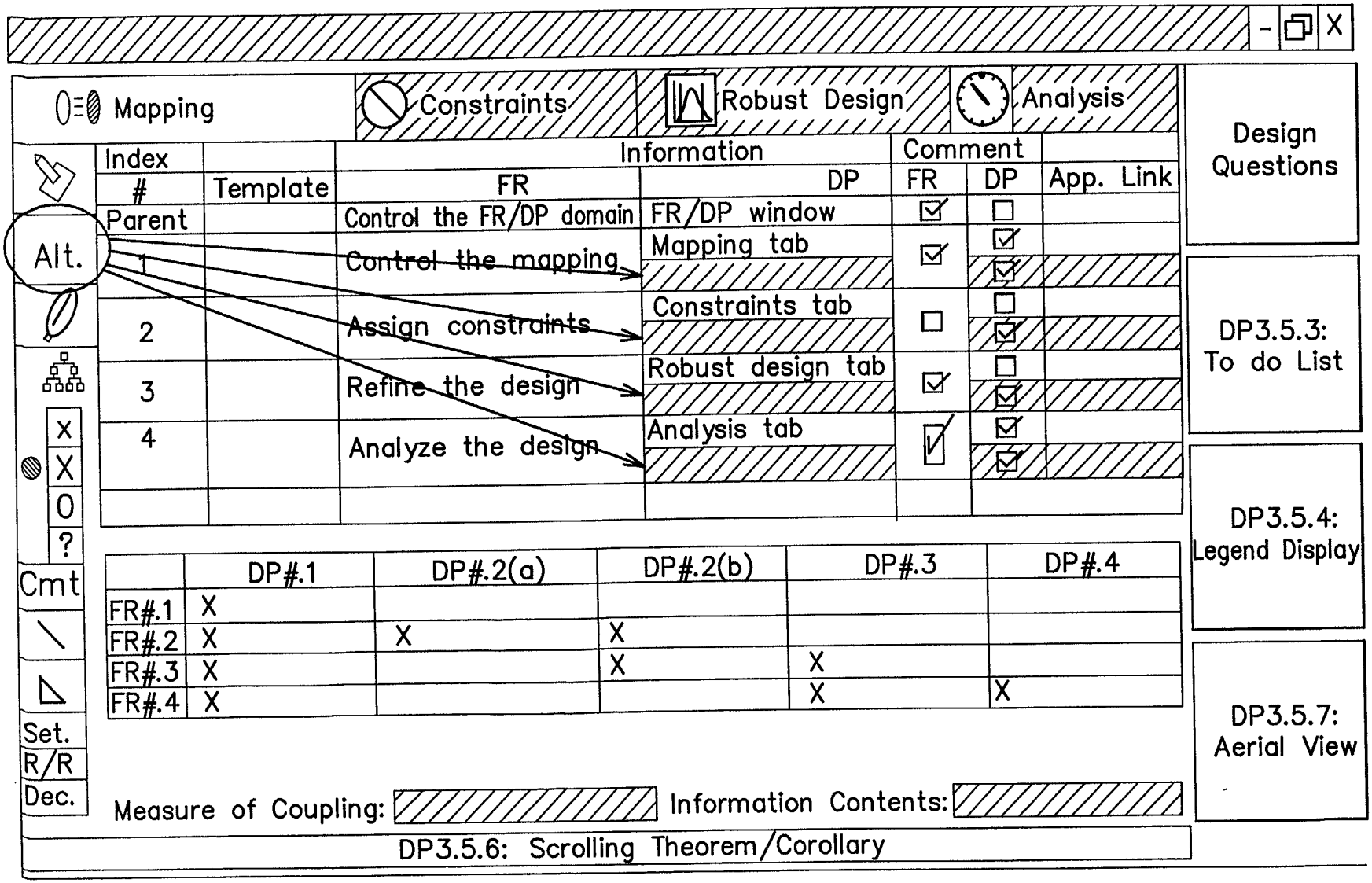


FIG. 58

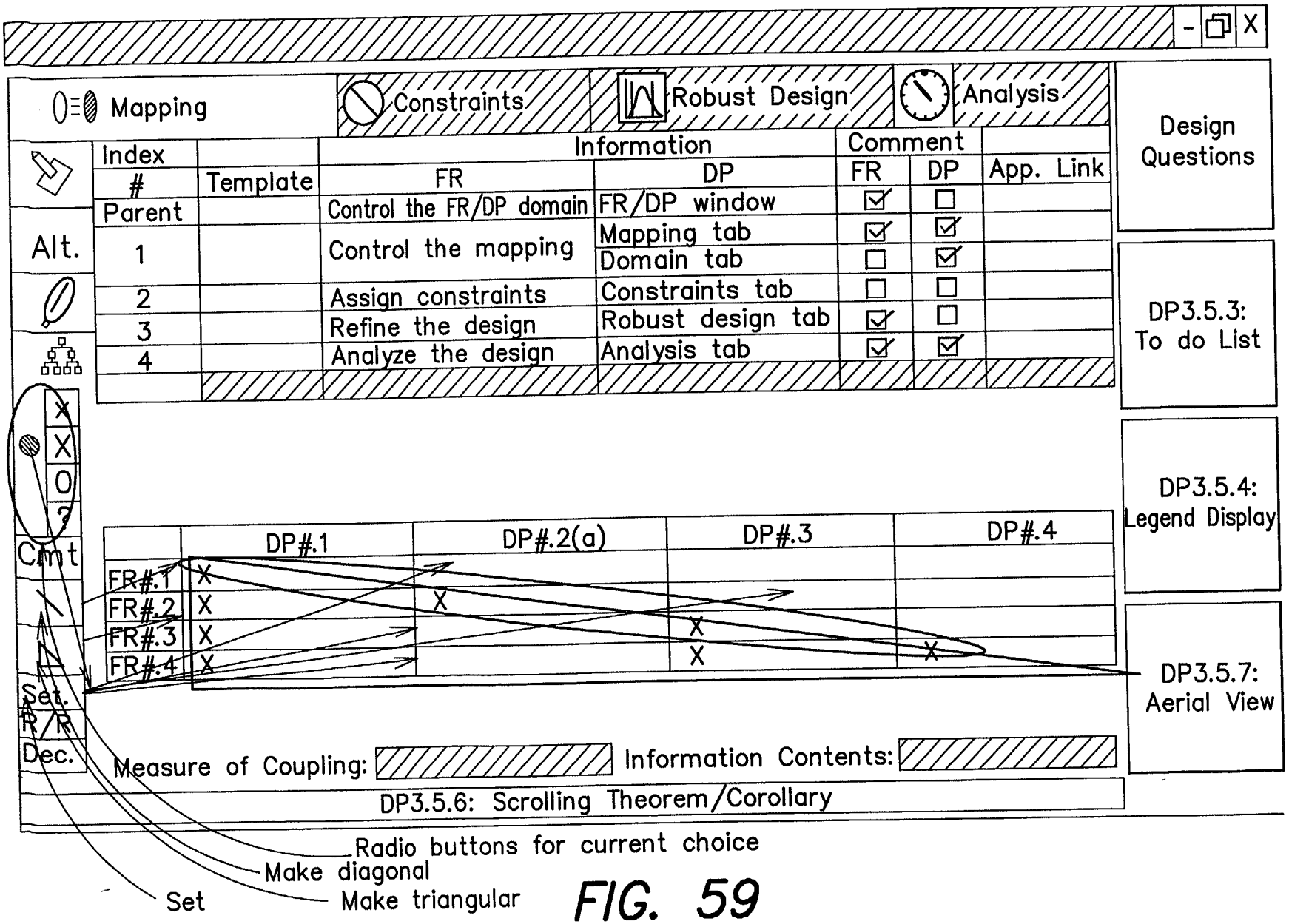

Mapping
Constraints
Robust Design
Analysis
Design Questions
Index
#
Template
Information
FR
DP
Comment
FR
DP
App. Link
Parent
Control the FR/DP domain
FR/DP window
Alt.
1
Control the mapping
Mapping tab
Domain tab
2
Assign constraints
Constraints tab
3
Refine the design
Robust design tab
4
Analyze the design
Analysis tab
DP3.5.3: To do List
DP3.5.4: Legend Display
DP3.5.7: Aerial View
Cmt
Set.
R/R
Dec.
DP#.1
DP#.2(a)
DP#.3
DP#.4
FR#.1
FR#.2
FR#.3
FR#.4
Measure of Coupling:
Information Contents:
DP3.5.6: Scrolling Theorem/Corollary
Radio buttons for current choice
Make diagonal
Make triangular
Set

FIG. 59

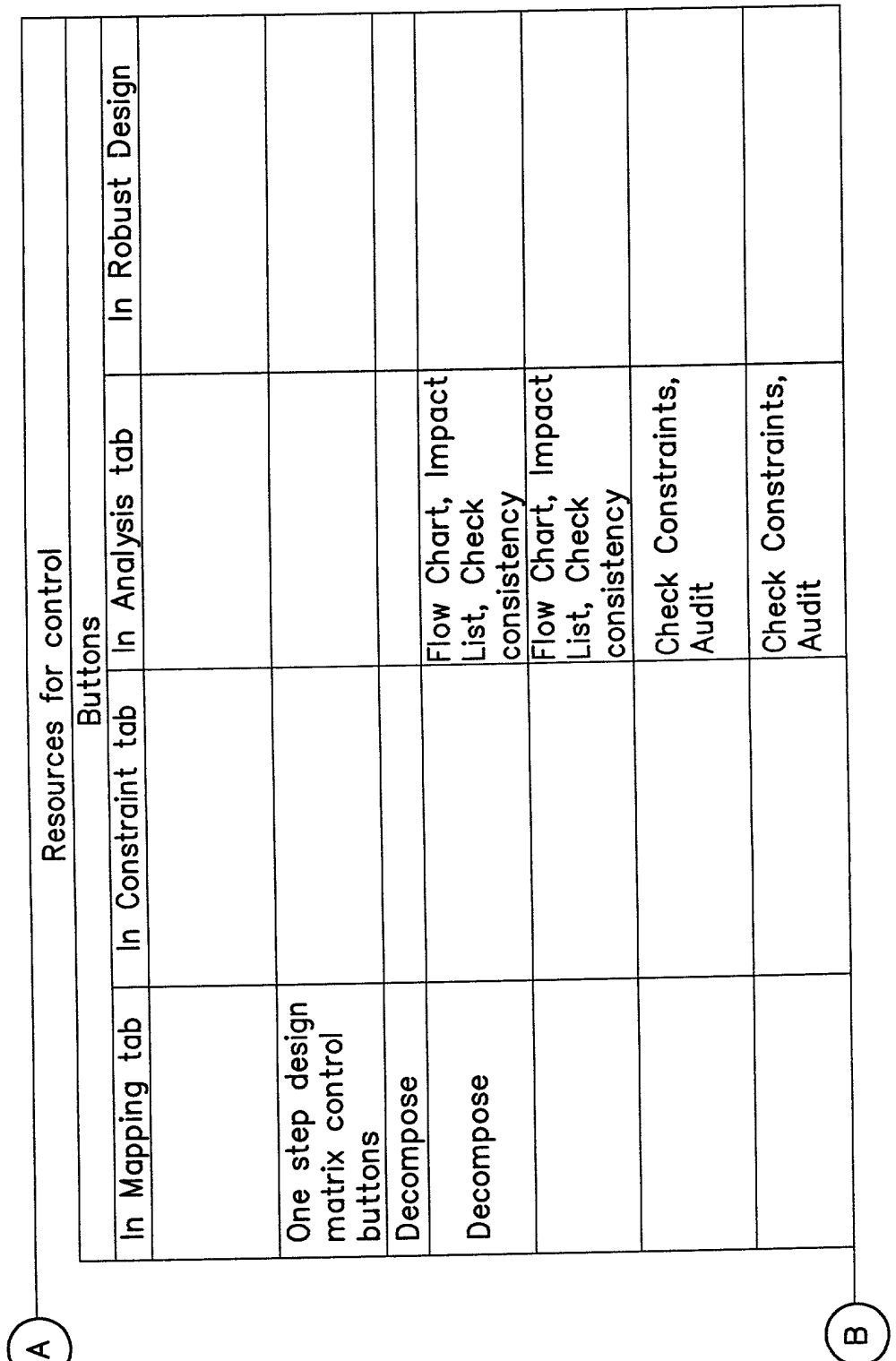
Roadmap		Is this step finished?		Resources for control		
		Yes	No	Menu	Tab	Toolbar
Start the design process		Disable		View–> Project Control	Constraints, Robust design, Analysis	Project Control
Activities at one level of the design hierarchy	FR/DP mapping	Enable			Constraints	
			Disable			
	Define Design Matrix	Enable			Analysis	
			Disable			
Activities over the design hierarchy	Define leaf level	Enable		View–> Project Control	Robust Design	Project Control
			Disable			

A

B

FIG. 60A | FIG. 60B

FIG. 60A



Resources for control			
Buttons			
In Mapping tab	In Constraint tab	In Analysis tab	In Robust Design
One step design matrix control buttons			
Decompose			
Decompose		Flow Chart, Impact List, Check consistency	
		Flow Chart, Impact List, Check consistency	
		Check Constraints, Audit	
		Check Constraints, Audit	

FIG. 60B

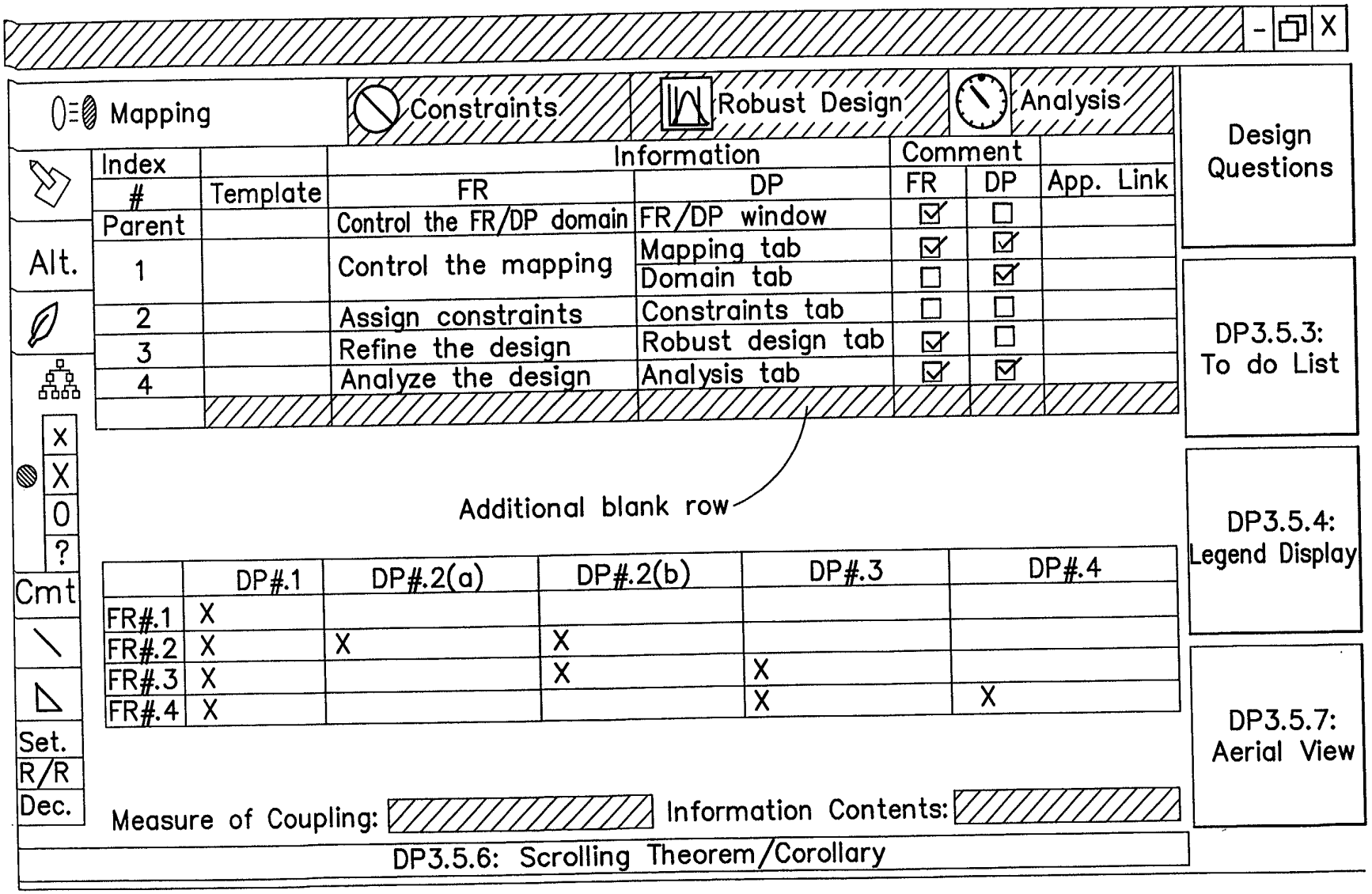

Mapping
Constraints
Robust Design
Analysis
Design Questions
Index
#
Template
Information
FR
DP
Comment
FR
DP
App. Link
Parent
Control the FR/DP domain
FR/DP window
Alt.
1
Control the mapping
Mapping tab
Domain tab
2
Assign constraints
Constraints tab
3
Refine the design
Robust design tab
4
Analyze the design
Analysis tab
DP3.5.3: To do List
Additional blank row
X
X
O
?
Cmt
Set.
R/R
Dec.
DP3.5.4: Legend Display
DP#.1
DP#.2(a)
DP#.2(b)
DP#.3
DP#.4
FR#.1
FR#.2
FR#.3
FR#.4
DP3.5.7: Aerial View
Measure of Coupling:
Information Contents:
DP3.5.6: Scrolling Theorem/Corollary

FIG. 61

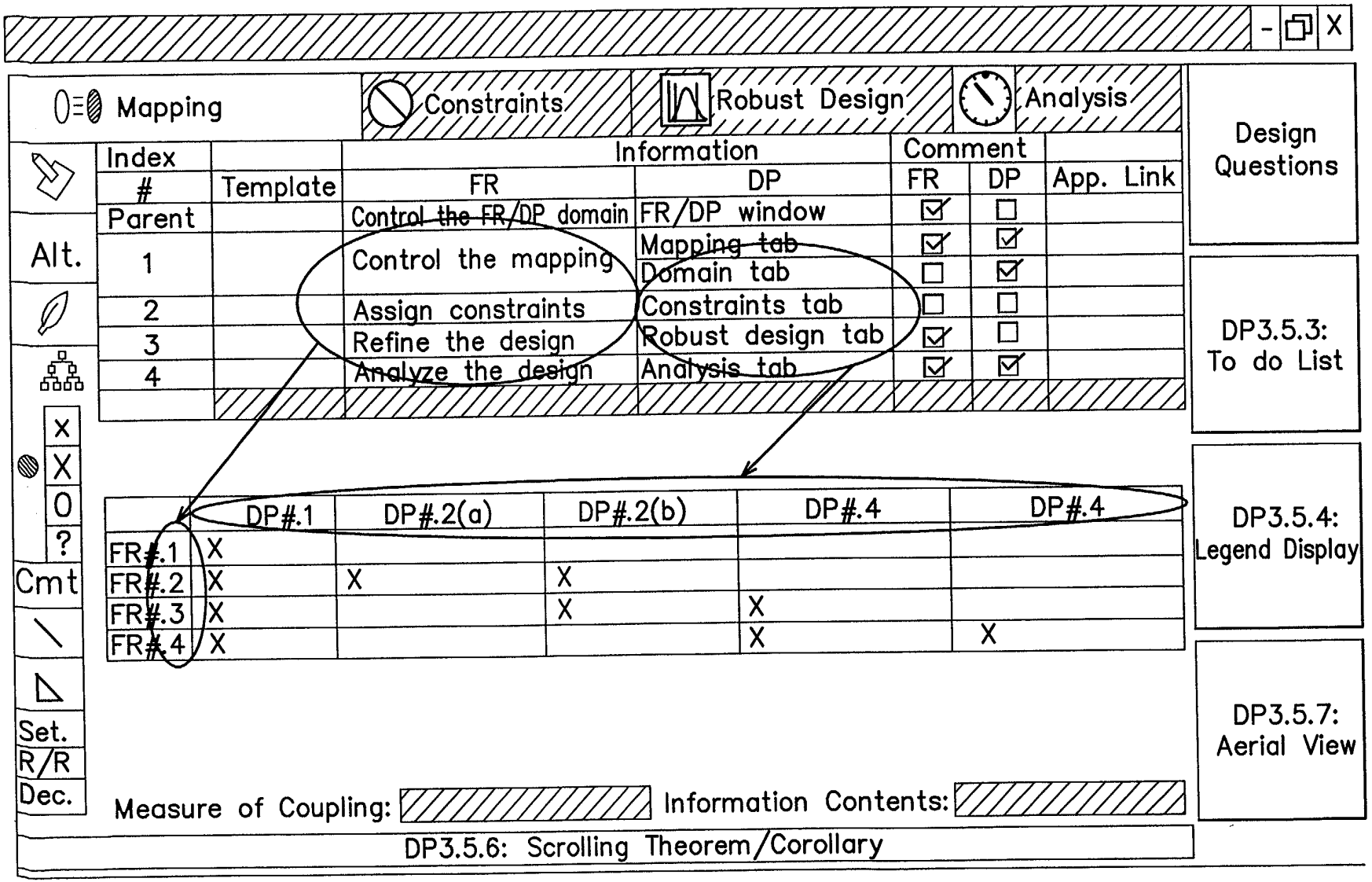

Mapping
Constraints
Robust Design
Analysis
Design Questions
DP3.5.3: To do List
DP3.5.4: Legend Display
DP3.5.7: Aerial View
Index
#
Parent
Template
Information
FR
DP
Comment
FR
DP
App. Link
Control the FR/DP domain
FR/DP window
1
Control the mapping
Mapping tab
Domain tab
2
Assign constraints
Constraints tab
3
Refine the design
Robust design tab
4
Analyze the design
Analysis tab
Alt.
Cmt
Set.
R/R
Dec.
X
X
O
?
DP#.1
DP#.2(a)
DP#.2(b)
DP#.4
DP#.4
FR#.1
FR#.2
FR#.3
FR#.4
Measure of Coupling:
Information Contents:
DP3.5.6: Scrolling Theorem/Corollary

FIG. 62

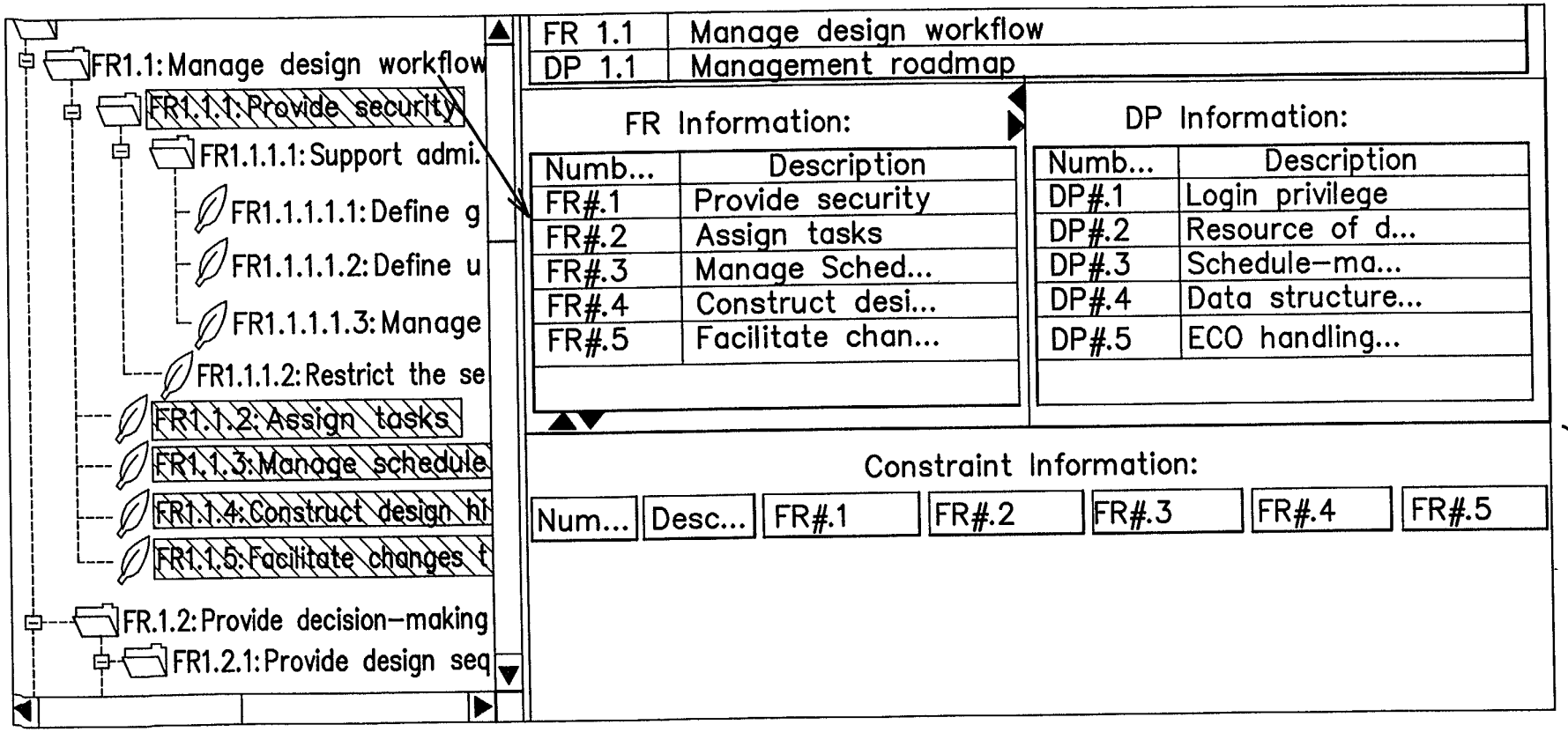

FR1.1:Manage design workflow
FR1.1.1:Provide security
FR1.1.1.1:Support admi.
FR1.1.1.1.1:Define g
FR1.1.1.1.2:Define u
FR1.1.1.1.3:Manage
FR1.1.1.2:Restrict the se
FR1.1.2:Assign tasks
FR1.1.3:Manage schedule
FR1.1.4:Construct design hi
FR1.1.5:Facilitate changes t
FR.1.2:Provide decision-making
FR1.2.1:Provide design seq
FR 1.1
Manage design workflow
DP 1.1
Management roadmap
FR Information:
Numb...
Description
FR#.1
Provide security
FR#.2
Assign tasks
FR#.3
Manage Sched...
FR#.4
Construct desi...
FR#.5
Facilitate chan...
DP Information:
Numb...
Description
DP#.1
Login privilege
DP#.2
Resource of d...
DP#.3
Schedule-ma...
DP#.4
Data structure...
DP#.5
ECO handling...
Constraint Information:
Num...
Desc...
FR#.1
FR#.2
FR#.3
FR#.4
FR#.5

FIG. 63A

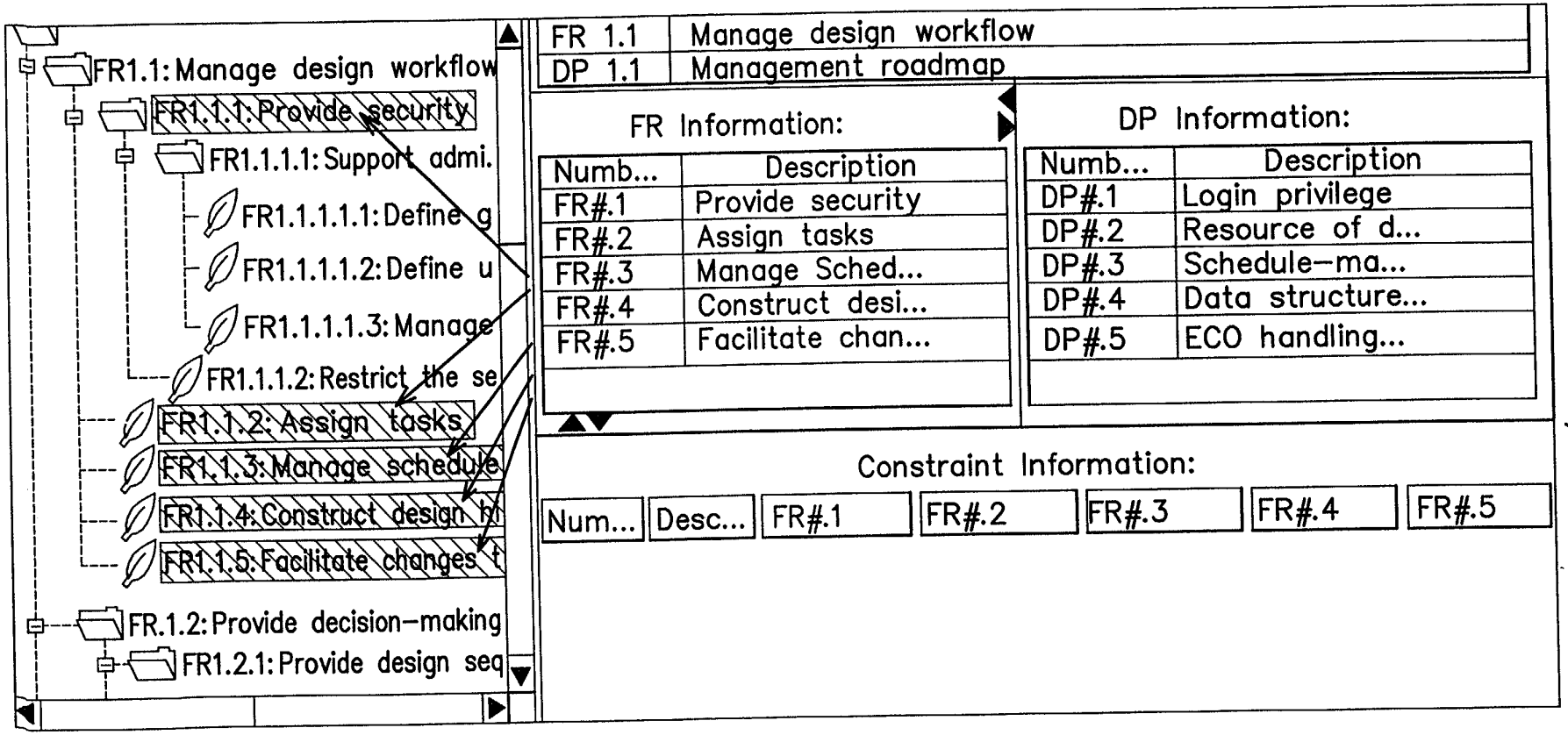


FIG. 63B

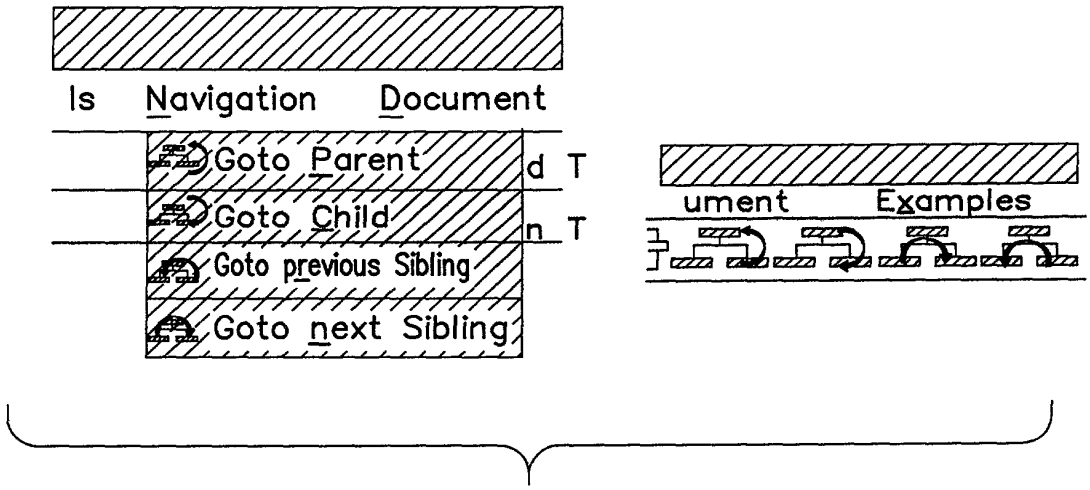

Is
Navigation
Document
Goto Parent
d T
Goto Child
n T
Goto previous Sibling
Goto next Sibling
ument
Examples

FIG. 64

Control Item		Level 1 Beginner	Level 2	Level 3 Intermediate	Level 4	Level 5 Expert
Available Features	FR/DP Mapping	●	●	●	●	●
	Design Matrix	●	●	●	●	●
	Alternative DP		●	●	●	●
	Analysis–Flow Chart		●	●	●	●
	Constraints			●	●	●
	Comments			●	●	●
	CN			●	●	●
	CN/FR Mapping			●	●	●
	Analysis–Child List			●	●	●
	Analysis–Impact List			●	●	●
	DP/PV Mapping				●	●
	Analysis–Check Consistency				●	●
	Analysis–Check Constraints				●	●
	Templates				●	●
	Verification				●	●
	Application Link				●	●
	Analysis–Audit					●
	Nested(Full) Matrix Handling					●
	Robust Design					●
	Project Control					●

A

B

FIG. 65A
FIG. 65B
FIG. 65C

FIG. 65A

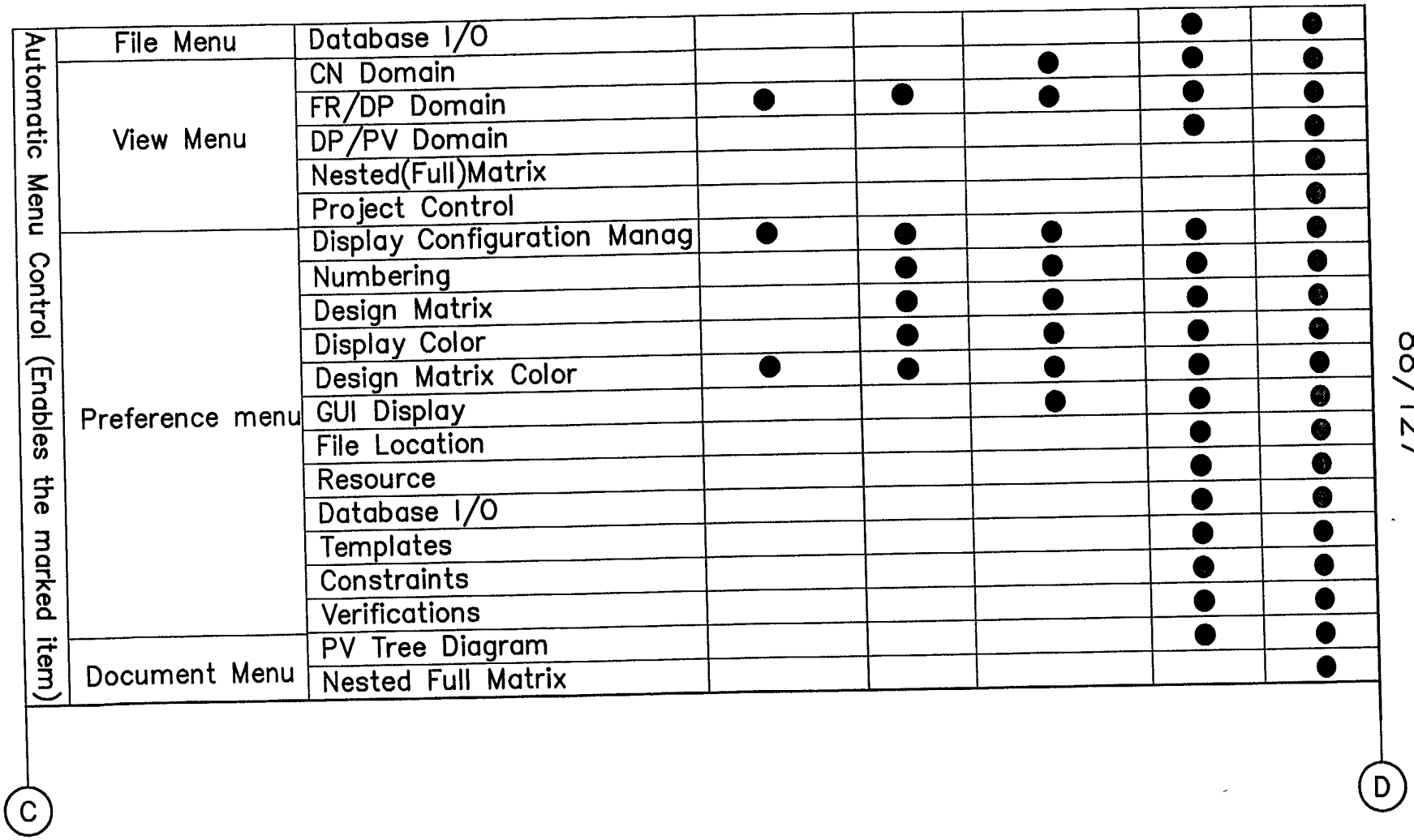

Automatic Menu Control (Enables the marked item)
File Menu
Database I/O
View Menu
CN Domain
FR/DP Domain
DP/PV Domain
Nested(Full)Matrix
Project Control
Preference menu
Display Configuration Manag
Numbering
Design Matrix
Display Color
Design Matrix Color
GUI Display
File Location
Resource
Database I/O
Templates
Constraints
Verifications
Document Menu
PV Tree Diagram
Nested Full Matrix
C
D

FIG. 65B

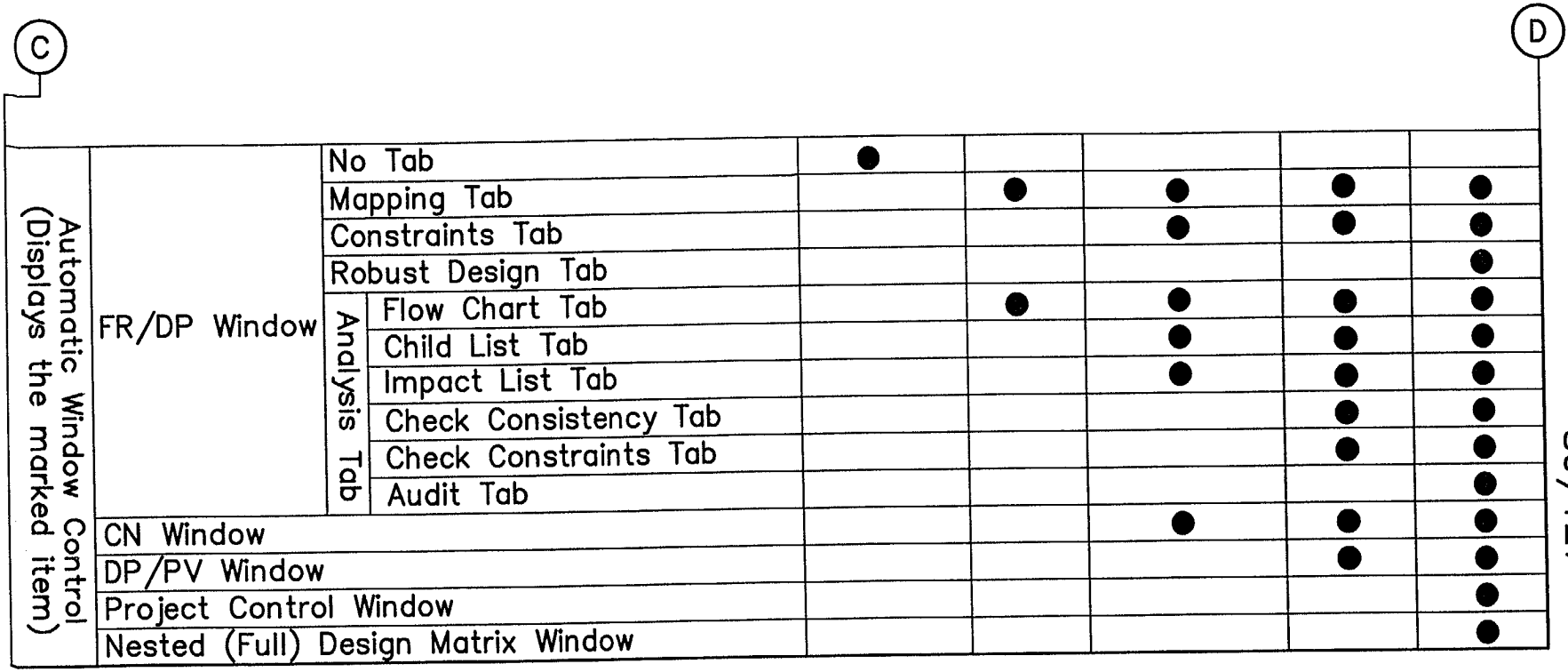

C
D
Automatic Window Control (Displays the marked item)
FR/DP Window
Analysis Tab
No Tab
Mapping Tab
Constraints Tab
Robust Design Tab
Flow Chart Tab
Child List Tab
Impact List Tab
Check Consistency Tab
Check Constraints Tab
Audit Tab
CN Window
DP/PV Window
Project Control Window
Nested (Full) Design Matrix Window

FIG. 65C

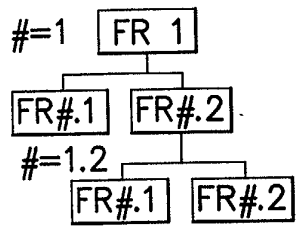
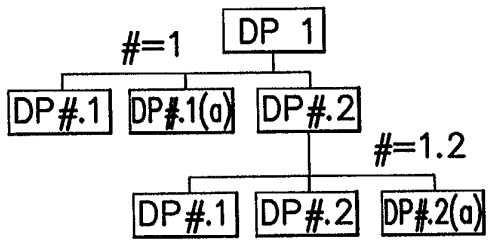
		Default Numbering	Alternative Numbering	Example
Numbering Type	Numeric	☑	☐	1, 2, 3
	Lower case	☐	☑	a, b, c
	Upper case	☐	☐	A, B, C
Indicator	Alternative connector	()		Defined by user
	Parent index	#		
	Divider			
Example				

FIG. 66

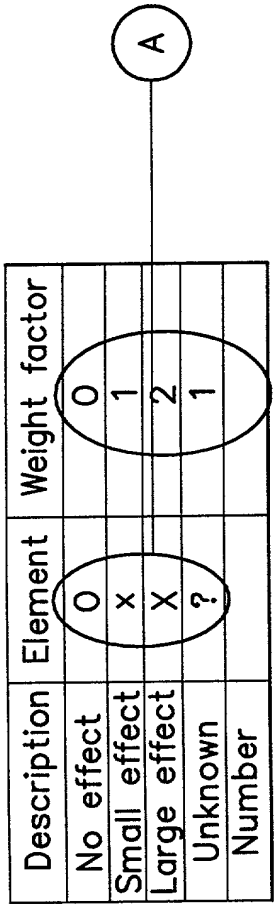


FIG. 67A | FIG. 67B

FIG. 67A

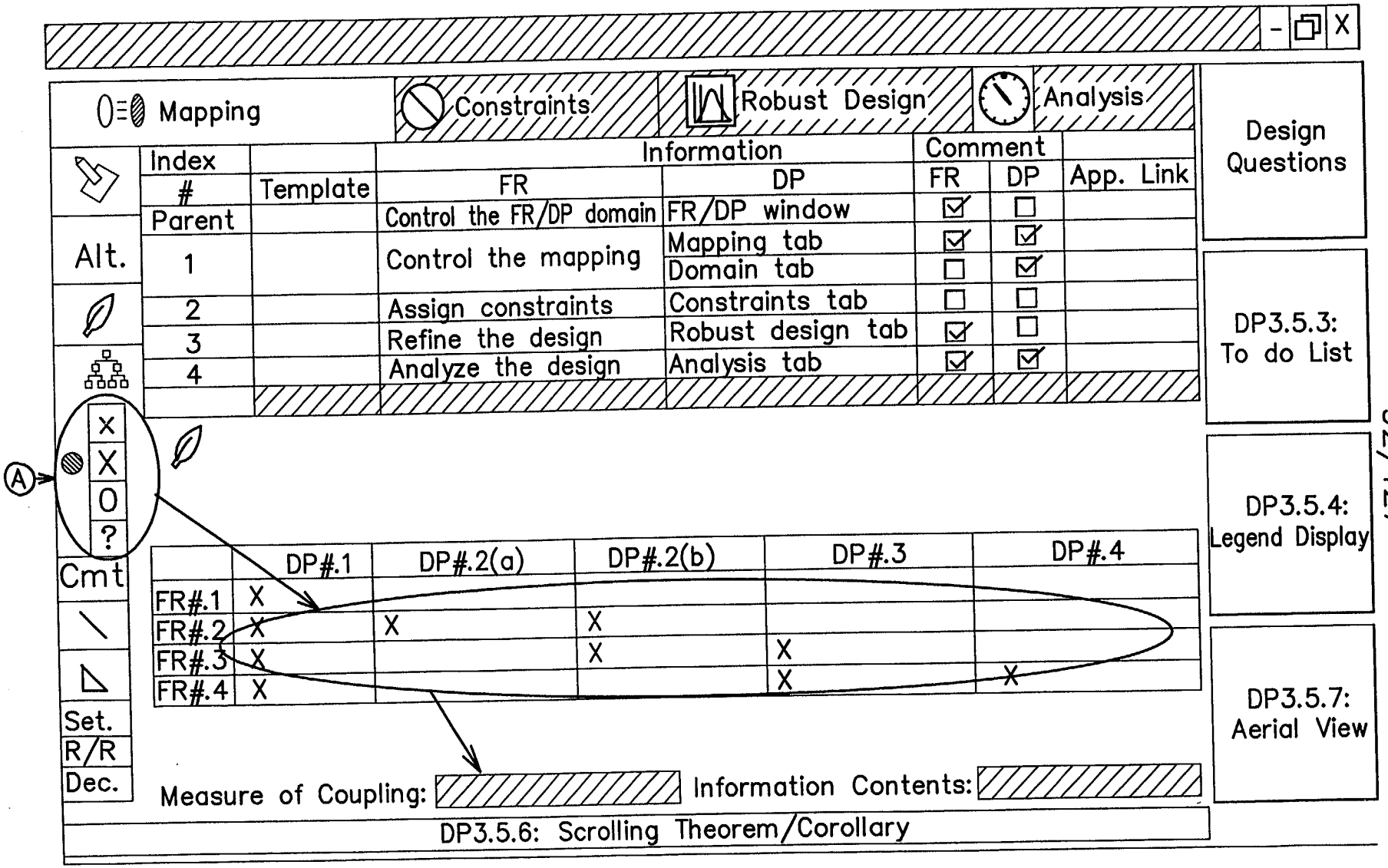

Mapping
Constraints
Robust Design
Analysis
Design Questions
Index
#
Parent
Template
Information
FR
DP
Comment
FR
DP
App. Link
Control the FR/DP domain
FR/DP window
Control the mapping
Mapping tab
Domain tab
Assign constraints
Constraints tab
Refine the design
Robust design tab
Analyze the design
Analysis tab
Alt.
Cmt
Set.
R/R
Dec.
X
X
O
?
A
DP#.1
DP#.2(a)
DP#.2(b)
DP#.3
DP#.4
FR#.1
FR#.2
FR#.3
FR#.4
Measure of Coupling:
Information Contents:
DP3.5.3: To do List
DP3.5.4: Legend Display
DP3.5.7: Aerial View
DP3.5.6: Scrolling Theorem/Corollary

FIG. 67B

		Legend category		
		Color	Font	Line
Display	Activated cell			N/A
	Normal			
	Default			N/A
	Focus			N/A
	Alternative			N/A
	Redundant			N/A
	Constraints			N/A
	Comments			N/A
Design Matrix	Uncoupled		N/A	
	Decoupled		N/A	
	Coupled		N/A	
	Undefined		N/A	
Template	Process			—
	Transport			---
	...			

FIG. 68

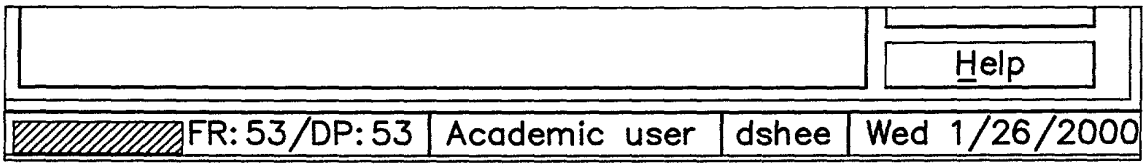


FIG. 69

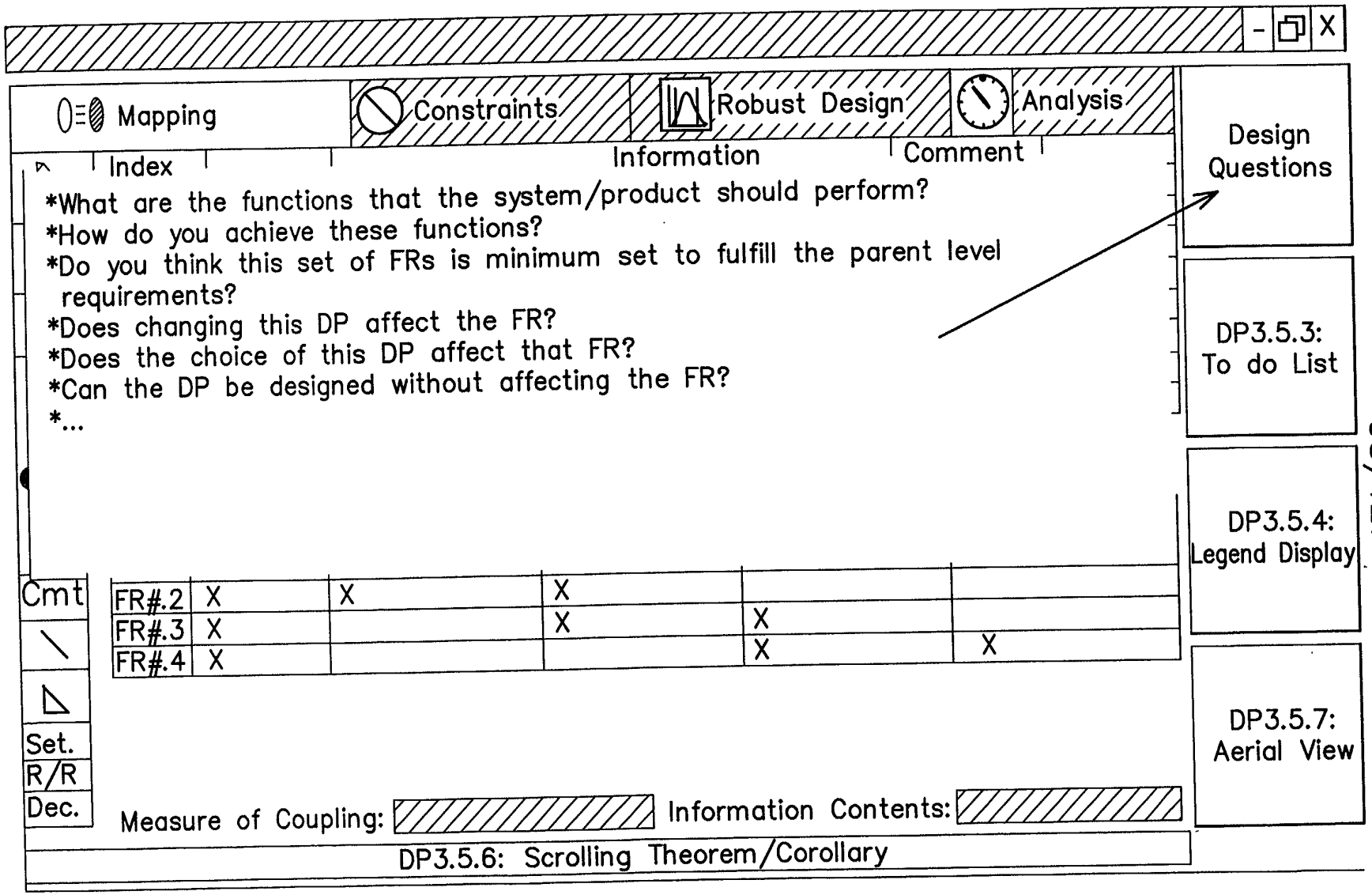

Mapping
Constraints
Robust Design
Analysis
Index
Information
Comment
*What are the functions that the system/product should perform?
*How do you achieve these functions?
*Do you think this set of FRs is minimum set to fulfill the parent level requirements?
*Does changing this DP affect the FR?
*Does the choice of this DP affect that FR?
*Can the DP be designed without affecting the FR?
*...
Design Questions
DP3.5.3: To do List
DP3.5.4: Legend Display
DP3.5.7: Aerial View
Cmt
Set.
R/R
Dec.
FR#.2 X X X
FR#.3 X X X
FR#.4 X X X
Measure of Coupling:
Information Contents:
DP3.5.6: Scrolling Theorem/Corollary

FIG. 70

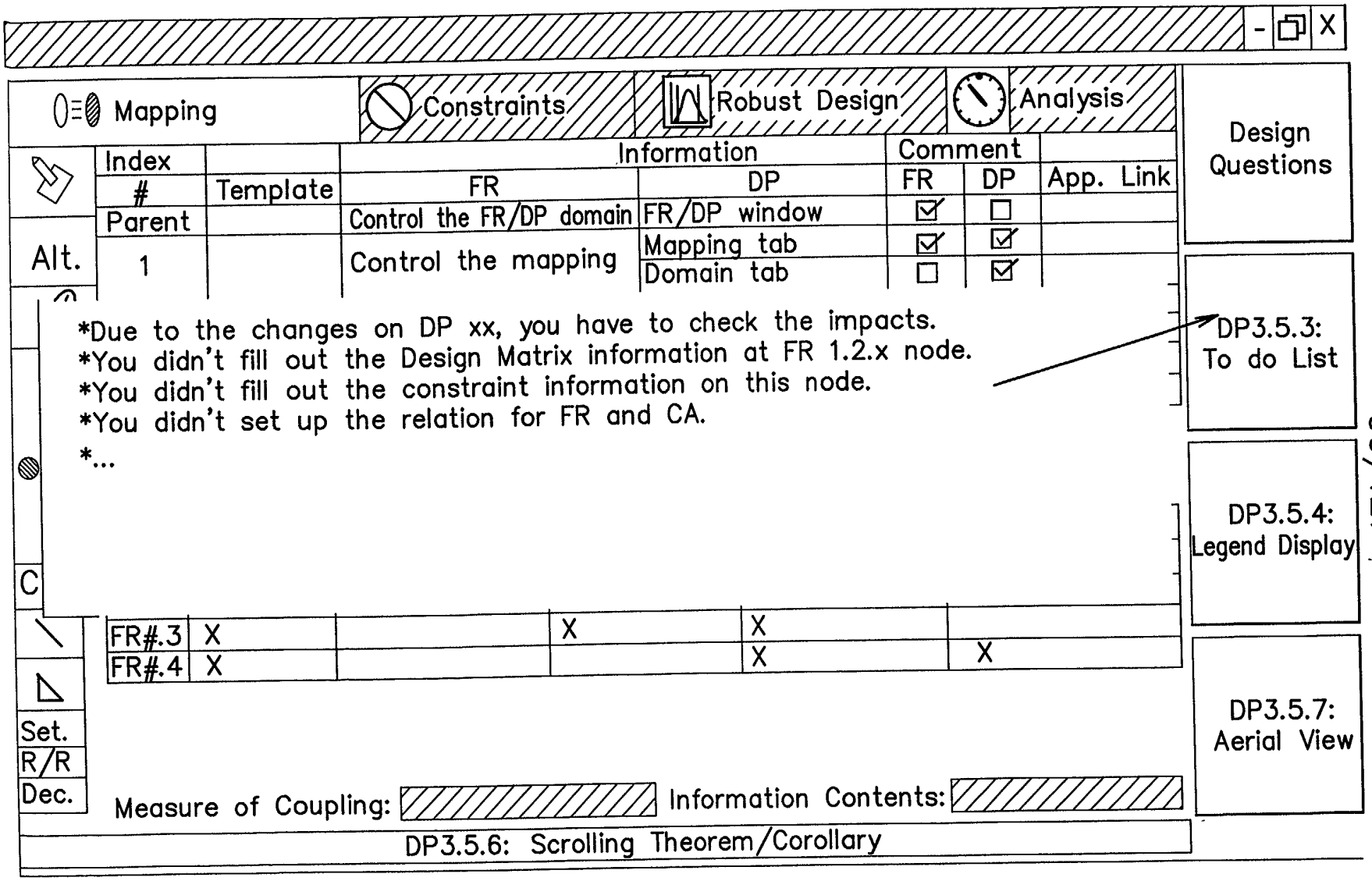

Mapping
Constraints
Robust Design
Analysis
Design Questions
Index
#
Template
Information
FR
DP
Comment
FR
DP
App. Link
Parent
Control the FR/DP domain
FR/DP window
Alt.
1
Control the mapping
Mapping tab
Domain tab
*Due to the changes on DP xx, you have to check the impacts.
*You didn't fill out the Design Matrix information at FR 1.2.x node.
*You didn't fill out the constraint information on this node.
*You didn't set up the relation for FR and CA.
*...
DP3.5.3: To do List
DP3.5.4: Legend Display
DP3.5.7: Aerial View
C
FR#.3
FR#.4
Set.
R/R
Dec.
Measure of Coupling:
Information Contents:
DP3.5.6: Scrolling Theorem/Corollary

FIG. 71

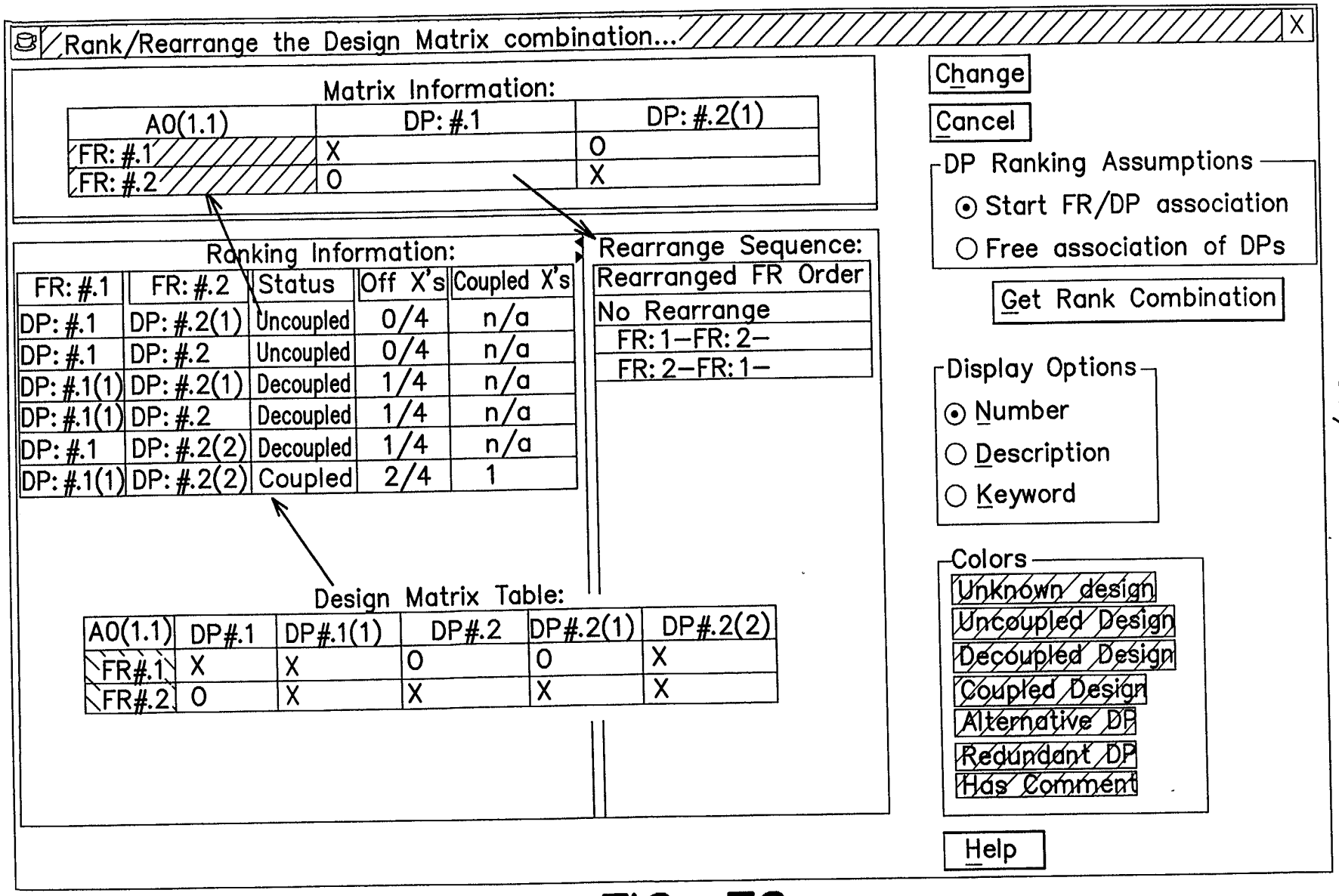

Rank/Rearrange the Design Matrix combination...
Matrix Information:
AO(1.1)
DP: #.1
DP: #.2(1)
FR: #.1
X
0
FR: #.2
0
X
Change
Cancel
DP Ranking Assumptions
Start FR/DP association
Free association of DPs
Get Rank Combination
Ranking Information:
FR: #.1
FR: #.2
Status
Off X's
Coupled X's
DP: #.1
DP: #.2(1)
Uncoupled
0/4
n/a
DP: #.1
DP: #.2
Uncoupled
0/4
n/a
DP: #.1(1)
DP: #.2(1)
Decoupled
1/4
n/a
DP: #.1(1)
DP: #.2
Decoupled
1/4
n/a
DP: #.1
DP: #.2(2)
Decoupled
1/4
n/a
DP: #.1(1)
DP: #.2(2)
Coupled
2/4
1
Rearrange Sequence:
Rearranged FR Order
No Rearrange
FR: 1–FR: 2–
FR: 2–FR: 1–
Display Options
Number
Description
Keyword
Colors
Unknown design
Uncoupled Design
Decoupled Design
Coupled Design
Alternative DP
Redundant DP
Has Comment
Design Matrix Table:
AO(1.1)
DP#.1
DP#.1(1)
DP#.2
DP#.2(1)
DP#.2(2)
FR#.1
X
X
0
0
X
FR#.2
0
X
X
X
X
Help

FIG. 72

Child List | Impact List | Inconsistency | Decoupling

Number	FR Description	DP Description
1.1	Manage design workflow	Management roadmap
1.1.1	Provide security	Login privilege
1.1.2	Assign tasks	Resource of design activity
1.1.3	Manage schedule	Schedule-managing tool (e.g. MS Project)
1.1.4	Construct design hierarchy	Data structure for Axiomatic Design concept
1.1.5	Facilitate changes to the design	ECO handling tool
1.1.1.1	Support administrative tool	User manager
1.1.1.2	Restrict the security access level	Authority code
1.1.1.1.1	Define group	Group specification
1.1.1.1.2	Define user	User specification
1.1.1.1.3	Manage authority code	Authority code specification

FIG. 73

Design Matrix Table:

A1(1.1)	DP#.1	DP#.2	DP#.3	DP#.4	DP#.5
FR#.1	X	0	0	0	0
FR#.2	X	X	0	0	0
FR#.3	X	X	X	X	X
FR#.4	X	X	0	X	0
FR#.5	0	0	0	X	X

Child List | Impact List | Inconsistency | Decoupling

Number	FR Description	DP Description
1.4.1	Support data file	File handling
1.4.2	Support database	Database handling
1.4.2.1	Provide consistency during data read a...	Data file format
1.4.2.2	Control error during read/write	Exception handling
1.4.2.3	Convert data from old version	Data file converter
1.4.2.4	Read data	Method for read
1.4.2.5	Write data	Method for write
1.4.2.6	Provide utility to deal with the program...	Method for utility
1.5	Provide utility function	Plug-in software
1.5.1	Handle external applications	Standard interface for external appli...
1.5.2	Teach the axiomatic design concept	Education software
1.5.3	Simulate the system architecture	Simulation software
1.5.4	Draw the Design Parameter figure	CAD Software
1.5.5	Analyze the system performance	Analysis software(i.e. ANSYS, NAS...
1.3	Support user friendliness of the software	Graphical User Interface software

Get data

Display Options
- ○ Number
- ○ Description
- ○ Keyword

Colors
- Uncoupled Design
- Decoupled Design
- Coupled Design
- No Effect
- Has Effect
- Has Comment

Help

FIG. 74

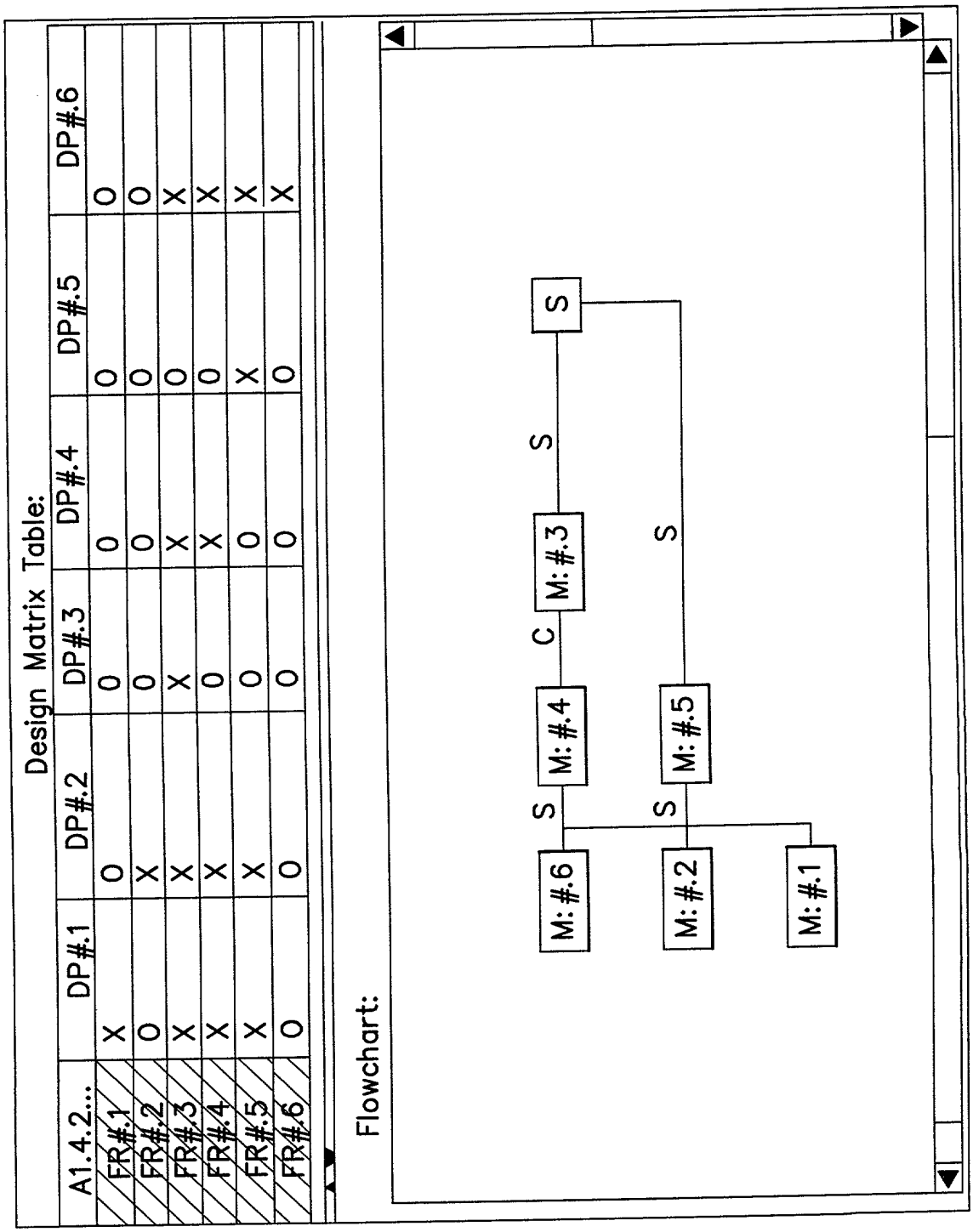

Design Matrix Table:
A1.4.2...
DP#.1
DP#.2
DP#.3
DP#.4
DP#.5
DP#.6
FR#.1
FR#.2
FR#.3
FR#.4
FR#.5
FR#.6
Flowchart:
M: #.6
M: #.4
M: #.3
M: #.2
M: #.5
M: #.1
S
C
S
S
S

FIG. 75

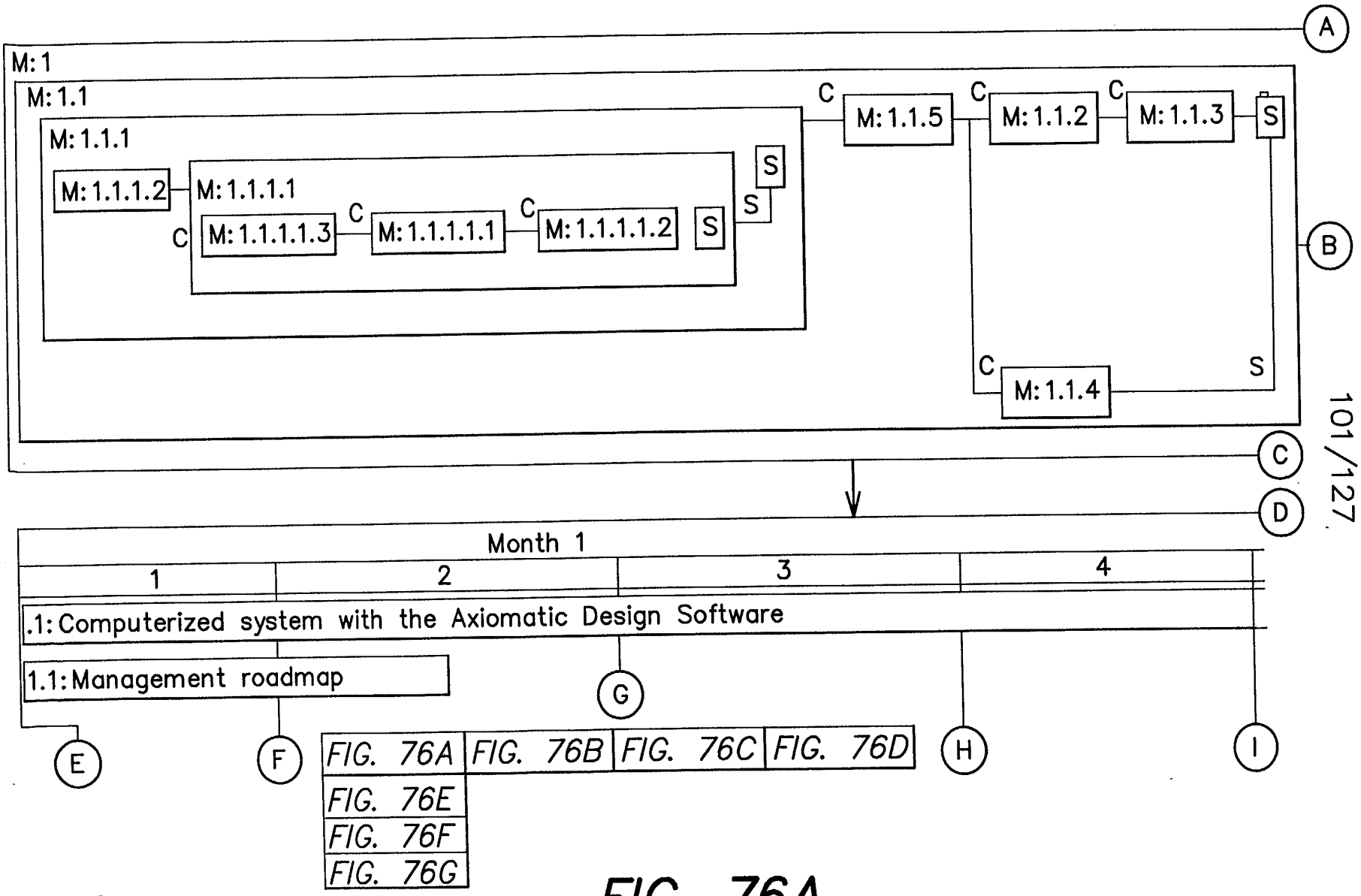

A
M: 1
M: 1.1
M: 1.1.1
M: 1.1.1.2
M: 1.1.1.1
M: 1.1.1.1.3
M: 1.1.1.1.1
M: 1.1.1.1.2
C
S
M: 1.1.5
M: 1.1.2
M: 1.1.3
M: 1.1.4
B
C
D
Month 1
1
2
3
4
.1: Computerized system with the Axiomatic Design Software
1.1: Management roadmap
E
F
G
H
I
FIG. 76A
FIG. 76B
FIG. 76C
FIG. 76D
FIG. 76E
FIG. 76F
FIG. 76G

FIG. 76A

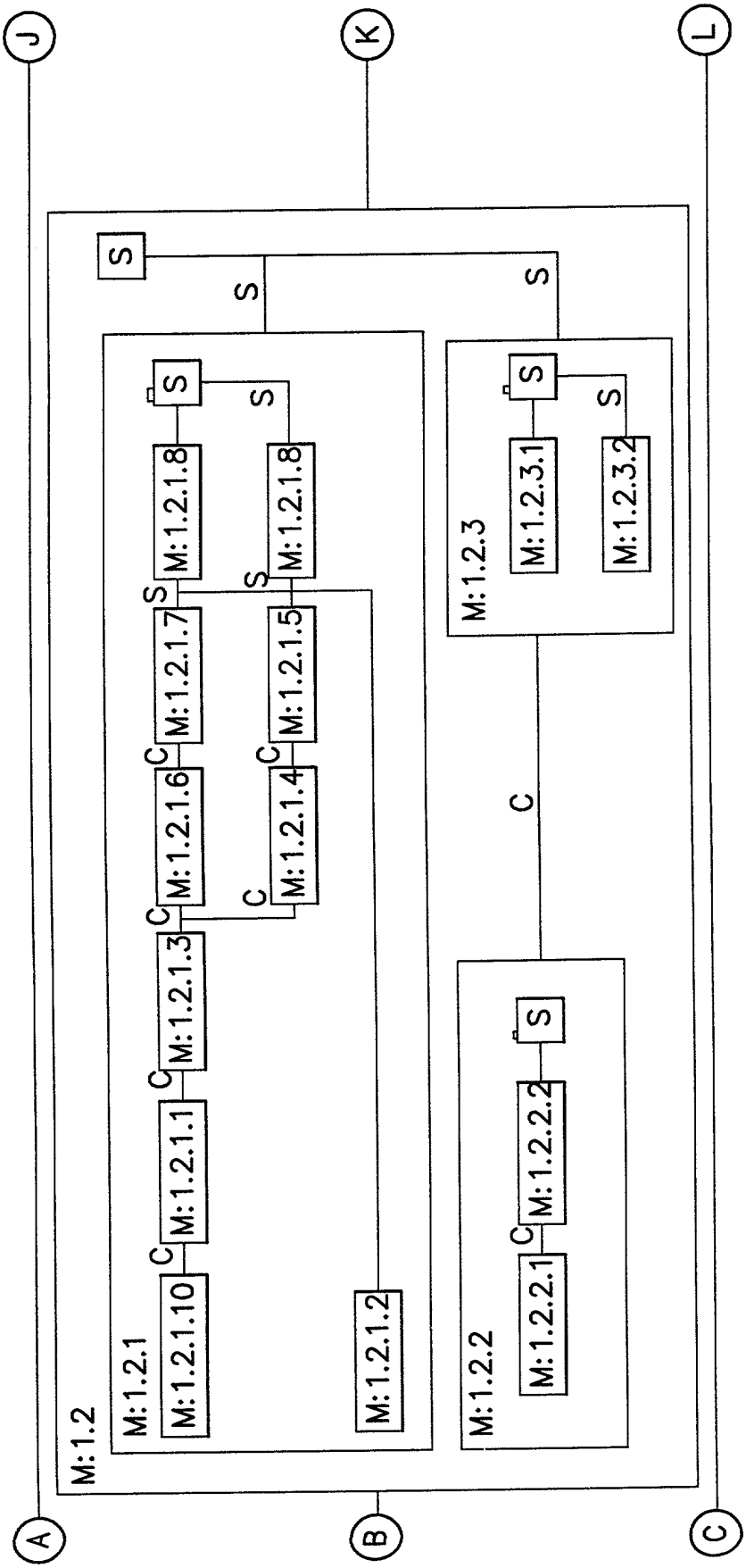

J
K
L
A
B
C
M:1.2
S
S
S
M:1.2.1
M:1.2.1.8
M:1.2.1.8
M:1.2.1.7
M:1.2.1.5
M:1.2.1.6
M:1.2.1.4
M:1.2.1.3
M:1.2.1.1
M:1.2.1.10
M:1.2.1.2
M:1.2.2
M:1.2.2.1
M:1.2.2.2
M:1.2.3
M:1.2.3.1
M:1.2.3.2
C

FIG. 76B

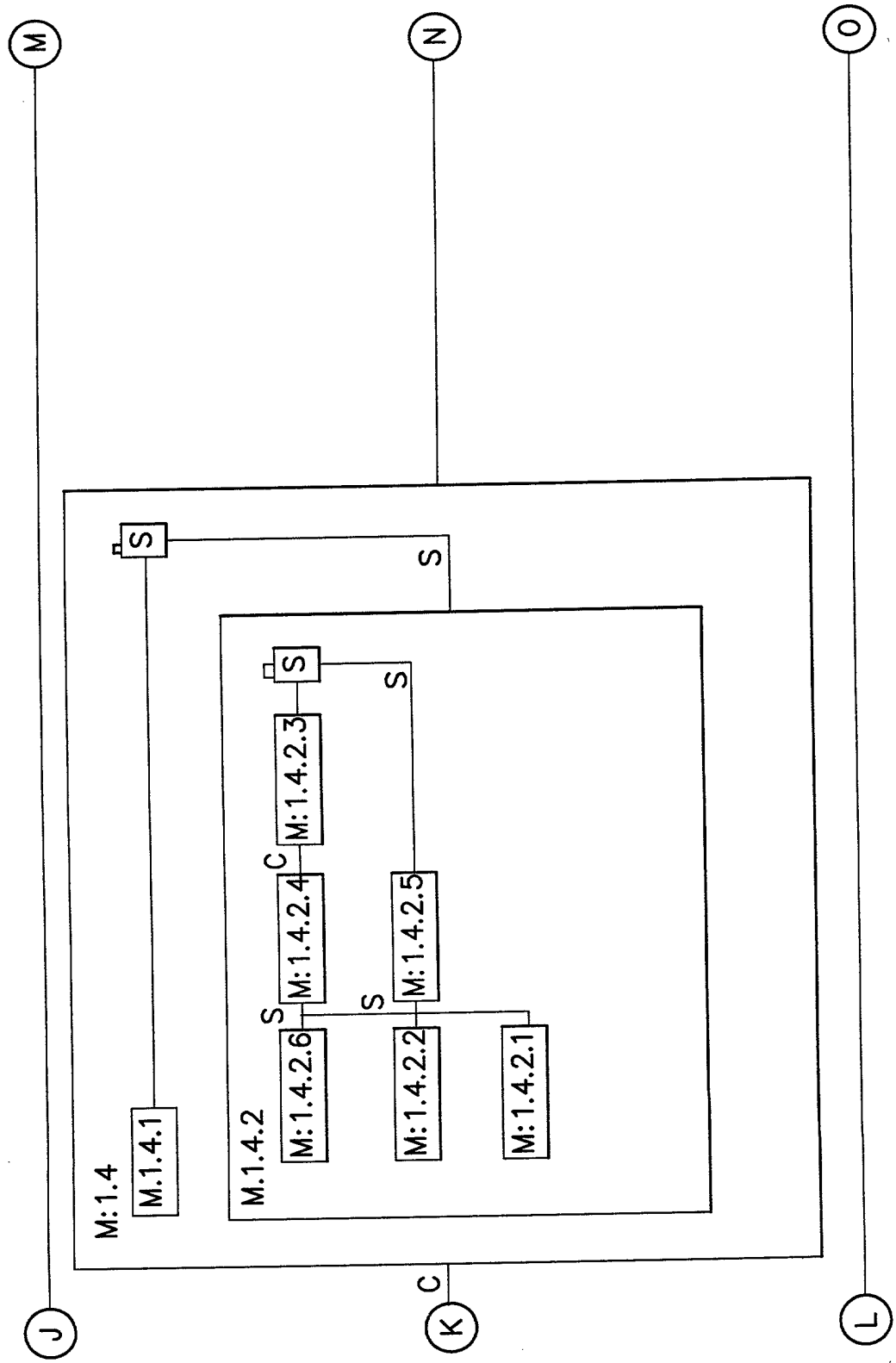

M
N
O
J
K
L
C
M:1.4
M.1.4.1
S
S
M.1.4.2
S
S
M:1.4.2.3
C
M:1.4.2.4
M:1.4.2.5
S
S
M:1.4.2.6
M:1.4.2.2
M:1.4.2.1

FIG. 76C

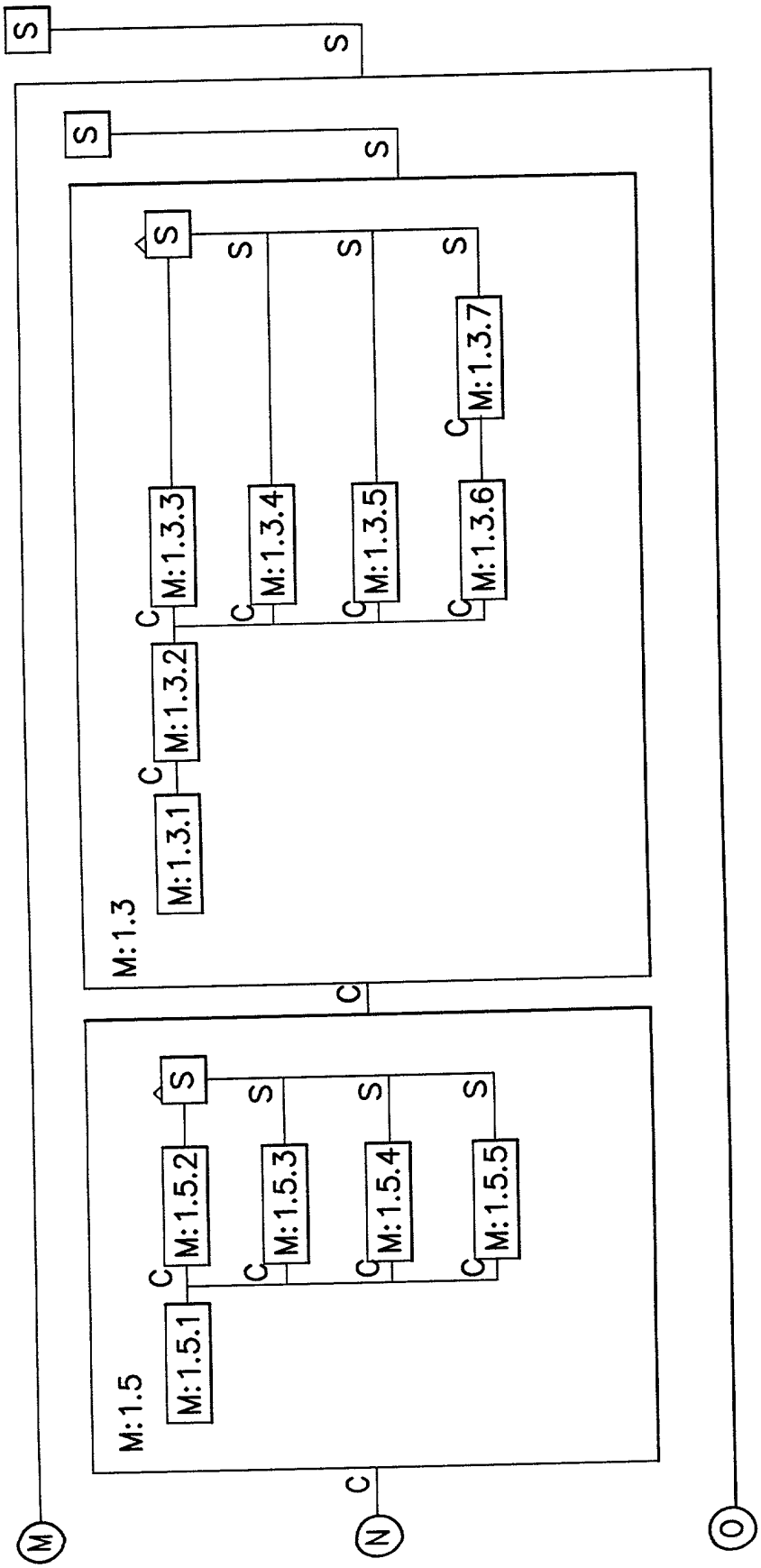

S
S
S
S
S
S
S
S
S
M:1.3.1
M:1.3.2
M:1.3.3
M:1.3.4
M:1.3.5
M:1.3.6
M:1.3.7
M:1.3
C
C
C
C
C
C
C
C
M:1.5.1
M:1.5.2
M:1.5.3
M:1.5.4
M:1.5.5
M:1.5
C
C
C
C
C
M
N
O

FIG. 76D

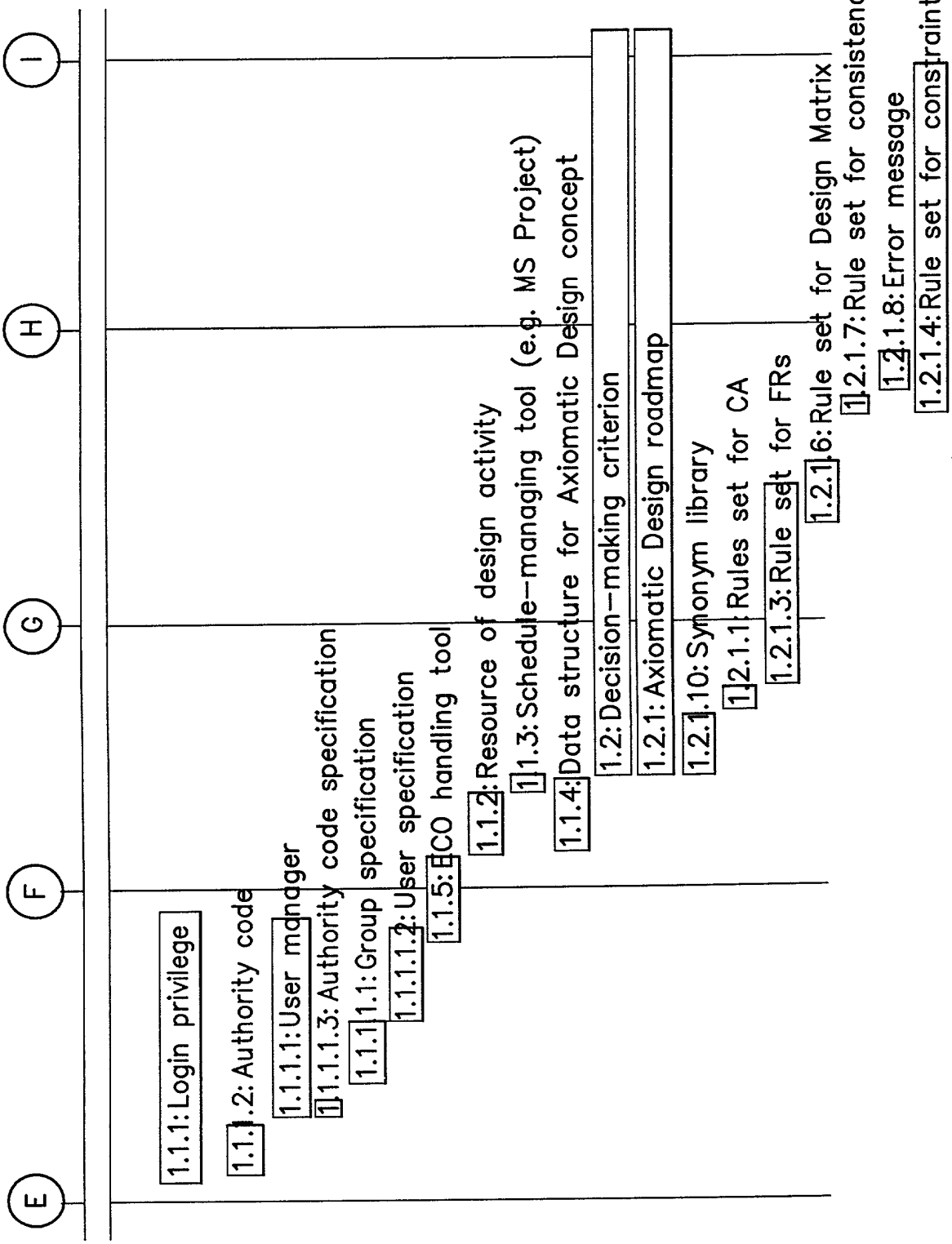

E
F
G
H
I
1.1.1:Login privilege
1.1.1.2:Authority code
1.1.1.1:User manager
1.1.1.3:Authority code specification
1.1.1 1.1:Group specification
1.1.1.1.2:User specification
1.1.5:ECO handling tool
1.1.2:Resource of design activity
1 1.3:Schedule-managing tool (e.g. MS Project)
1.1.4:Data structure for Axiomatic Design concept
1.2:Decision-making criterion
1.2.1:Axiomatic Design roadmap
1.2.1.10:Synonym library
1 2.1.1:Rules set for CA
1.2.1.3:Rule set for FRs
1.2.1.6:Rule set for Design Matrix
1 2.1.7:Rule set for consistency
1.2.1.8:Error message
1.2.1.4:Rule set for constraints

FIG. 76E

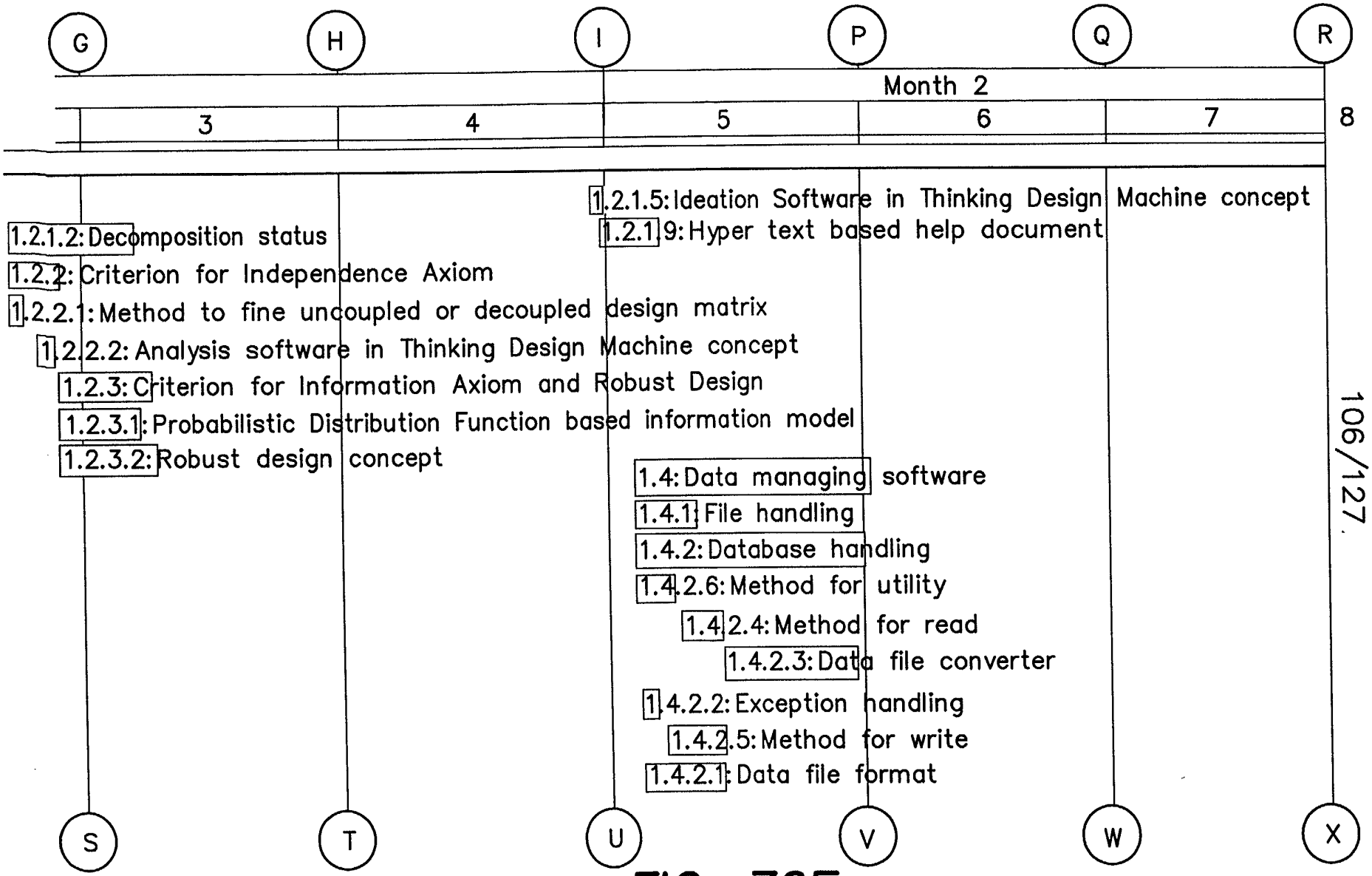


FIG. 76F

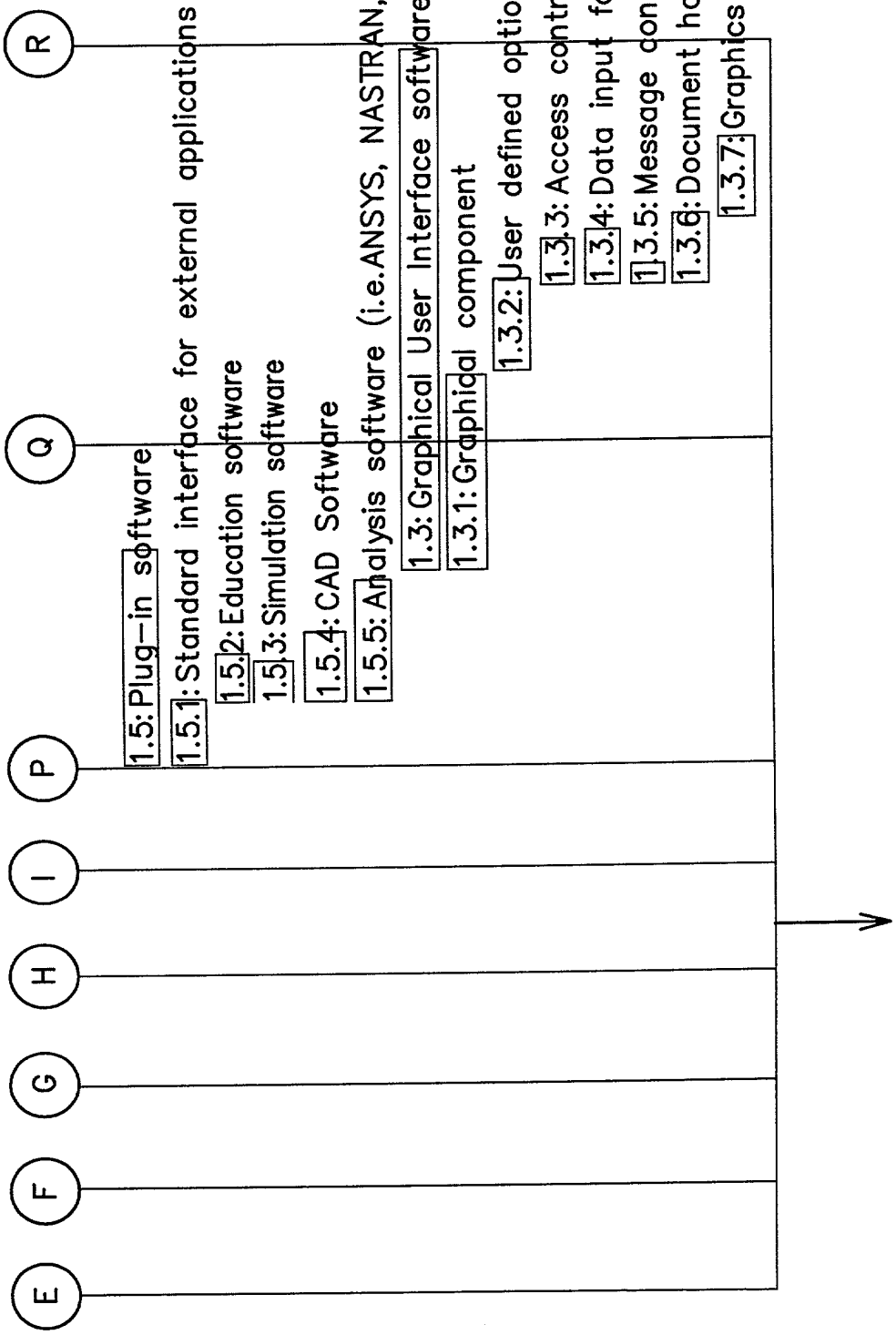

E
F
G
H
I
P
Q
R
1.5: Plug–in software
1.5.1: Standard interface for external applications
1.5.2: Education software
1.5.3: Simulation software
1.5.4: CAD Software
1.5.5: Analysis software (i.e.ANSYS, NASTRAN,..)
1.3: Graphical User Interface software
1.3.1: Graphidal component
1.3.2: User defined option
1.3.3: Access control
1.3.4: Data input format
1.3.5: Message control
1.3.6: Document handling
1.3.7: Graphics control method

FIG. 76G

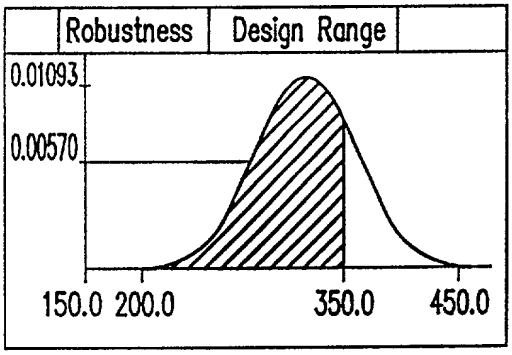


FIG. 77A

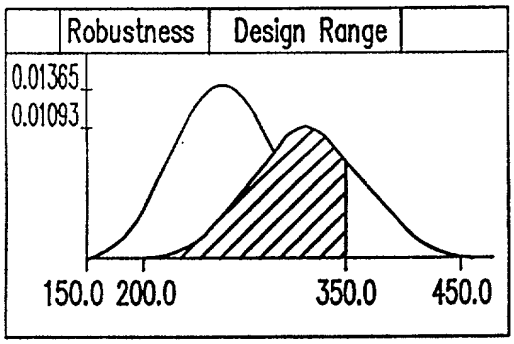


FIG. 77B

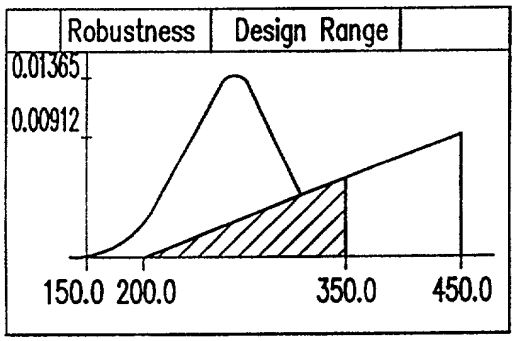


FIG. 77C

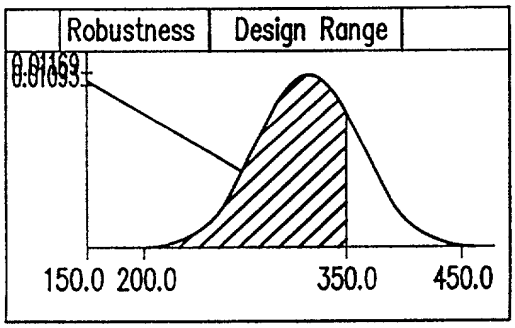


FIG. 77D

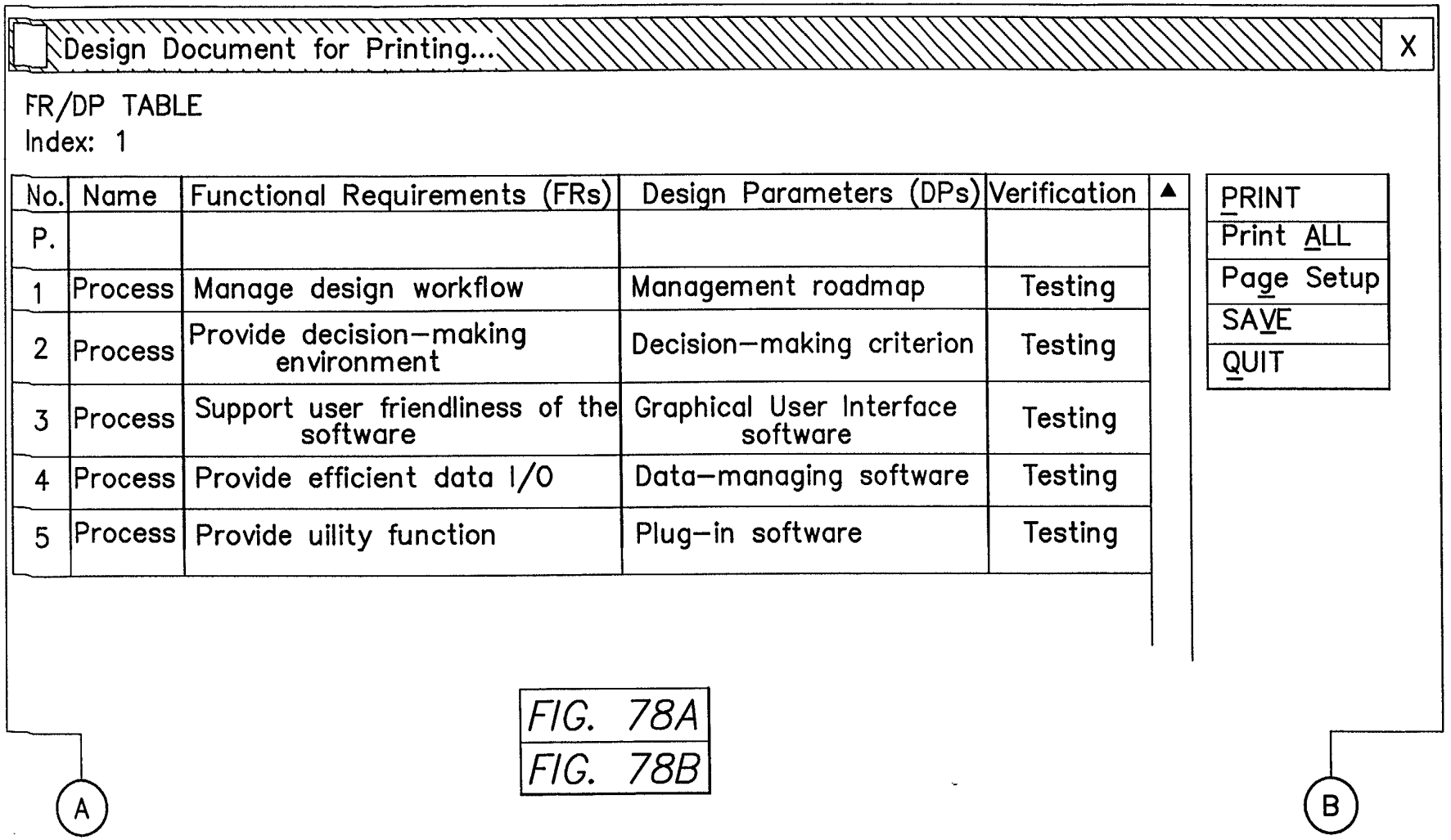


FIG. 78A
FIG. 78B

FIG. 78A

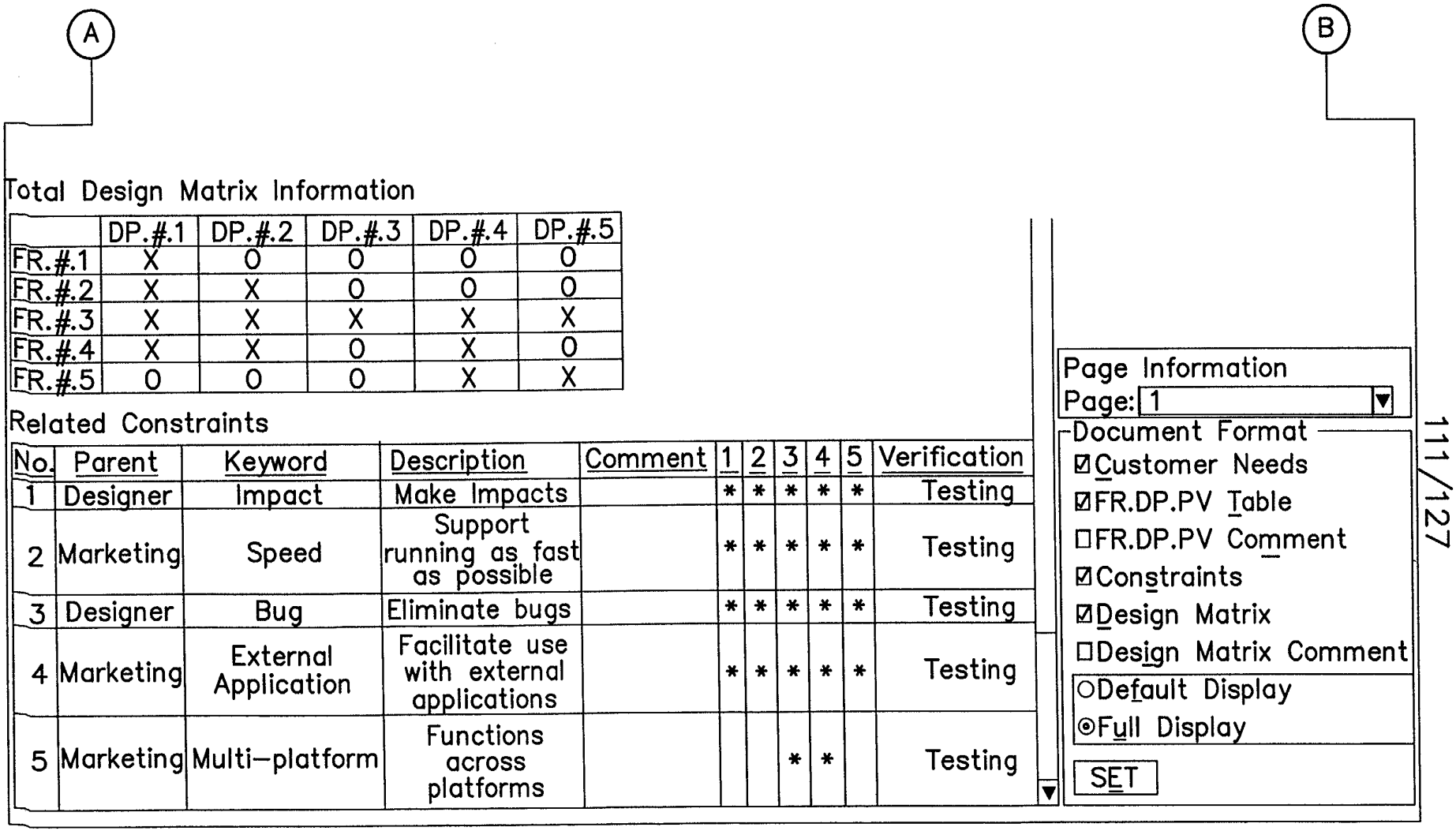

A
B
Total Design Matrix Information
DP.#.1 DP.#.2 DP.#.3 DP.#.4 DP.#.5
FR.#.1 X 0 0 0 0
FR.#.2 X X 0 0 0
FR.#.3 X X X X X
FR.#.4 X X 0 X 0
FR.#.5 0 0 0 X X
Related Constraints
No. Parent Keyword Description Comment 1 2 3 4 5 Verification
1 Designer Impact Make Impacts * * * * * Testing
2 Marketing Speed Support running as fast as possible * * * * * Testing
3 Designer Bug Eliminate bugs * * * * * Testing
4 Marketing External Application Facilitate use with external applications * * * * * Testing
5 Marketing Multi-platform Functions across platforms * * Testing
Page Information
Page: 1
Document Format
Customer Needs
FR.DP.PV Table
FR.DP.PV Comment
Constraints
Design Matrix
Design Matrix Comment
Default Display
Full Display
SET
111/127

FIG. 78B

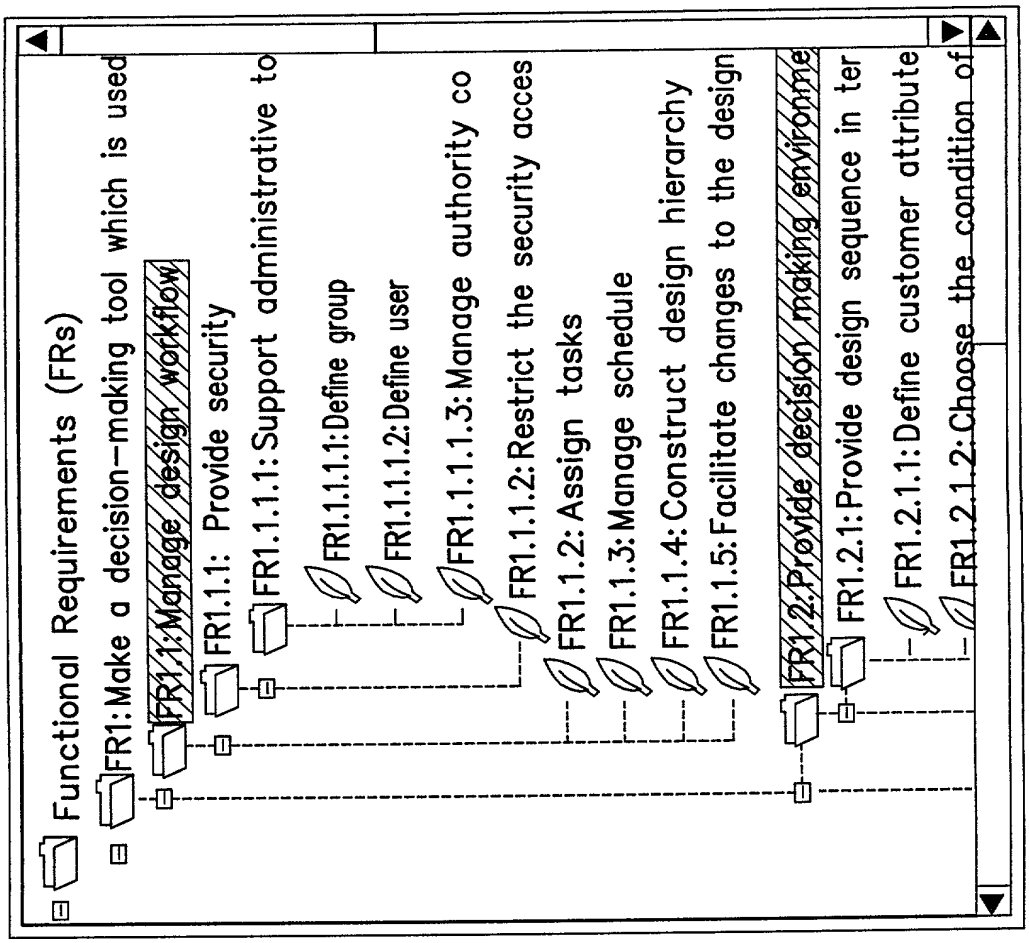

Functional Requirements (FRs)
FR1: Make a decision-making tool which is used
FR1.1: Manage design workflow
FR1.1.1: Provide security
FR1.1.1.1: Support administrative to
FR1.1.1.1.1: Define group
FR1.1.1.1.2: Define user
FR1.1.1.1.3: Manage authority co
FR1.1.1.2: Restrict the security acces
FR1.1.2: Assign tasks
FR1.1.3: Manage schedule
FR1.1.4: Construct design hierarchy
FR1.1.5: Facilitate changes to the design
FR1.2: Provide decision making environme
FR1.2.1: Provide design sequence in ter
FR1.2.1.1: Define customer attribute
FR1.2.1.2: Choose the condition of

FIG. 79A

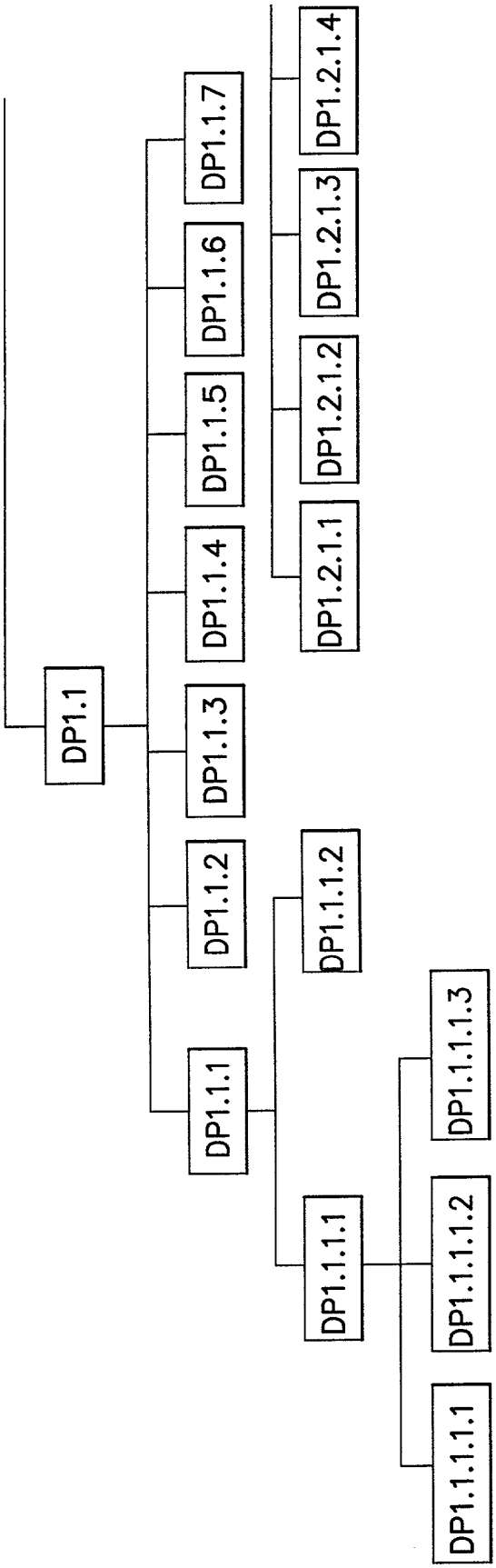

DP1.1
DP1.1.1
DP1.1.2
DP1.1.3
DP1.1.4
DP1.1.5
DP1.1.6
DP1.1.7
DP1.1.1.1
DP1.1.1.2
DP1.1.1.1.1
DP1.1.1.1.2
DP1.1.1.1.3
DP1.2.1.1
DP1.2.1.2
DP1.2.1.3
DP1.2.1.4

FIG. 79B

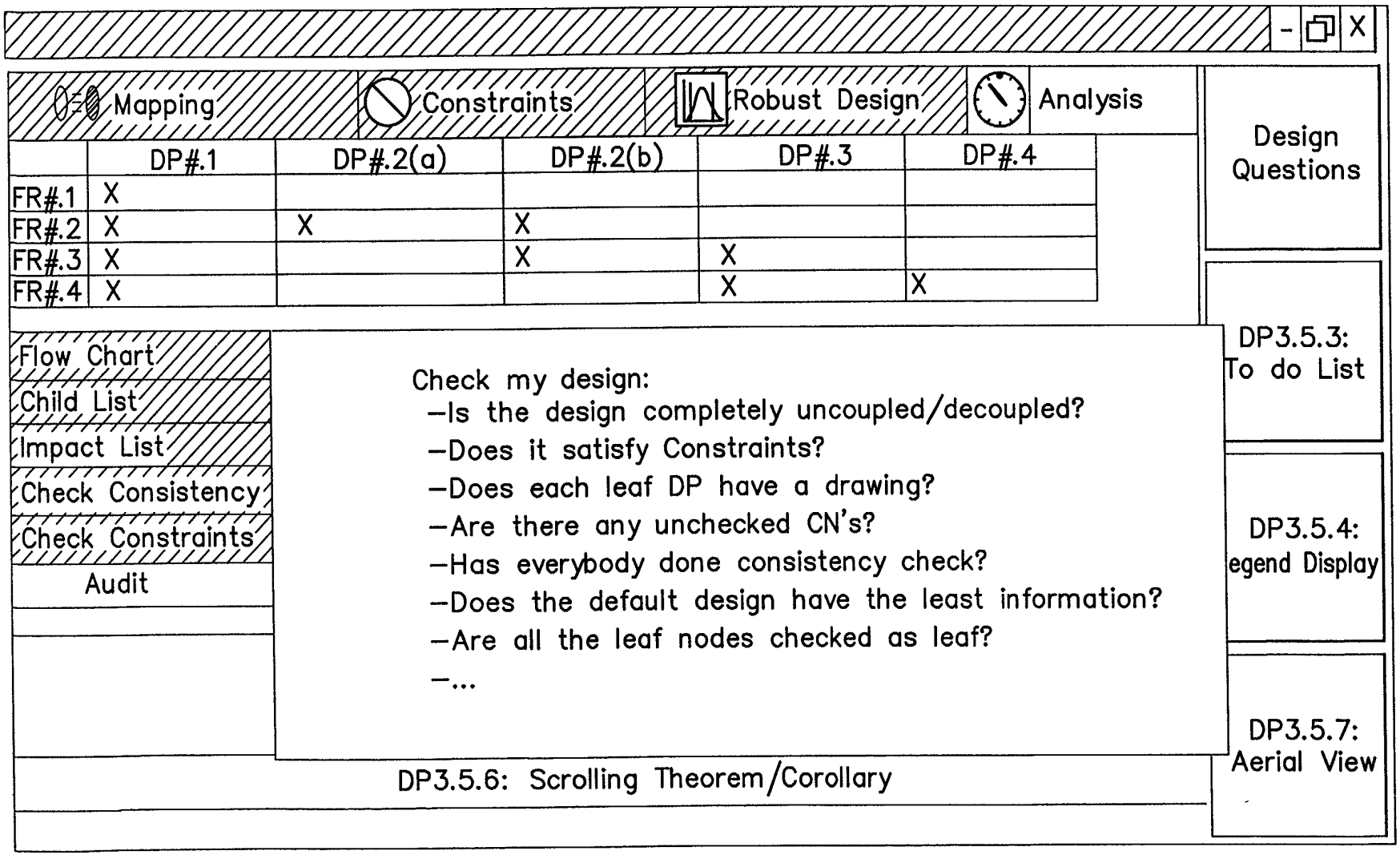

Mapping
Constraints
Robust Design
Analysis
DP#.1
DP#.2(a)
DP#.2(b)
DP#.3
DP#.4
FR#.1
FR#.2
FR#.3
FR#.4
Design Questions
Flow Chart
Child List
Impact List
Check Consistency
Check Constraints
Audit
Check my design:
–Is the design completely uncoupled/decoupled?
–Does it satisfy Constraints?
–Does each leaf DP have a drawing?
–Are there any unchecked CN's?
–Has everybody done consistency check?
–Does the default design have the least information?
–Are all the leaf nodes checked as leaf?
–...
DP3.5.3: To do List
DP3.5.4: egend Display
DP3.5.7: Aerial View
DP3.5.6: Scrolling Theorem/Corollary

FIG. 80

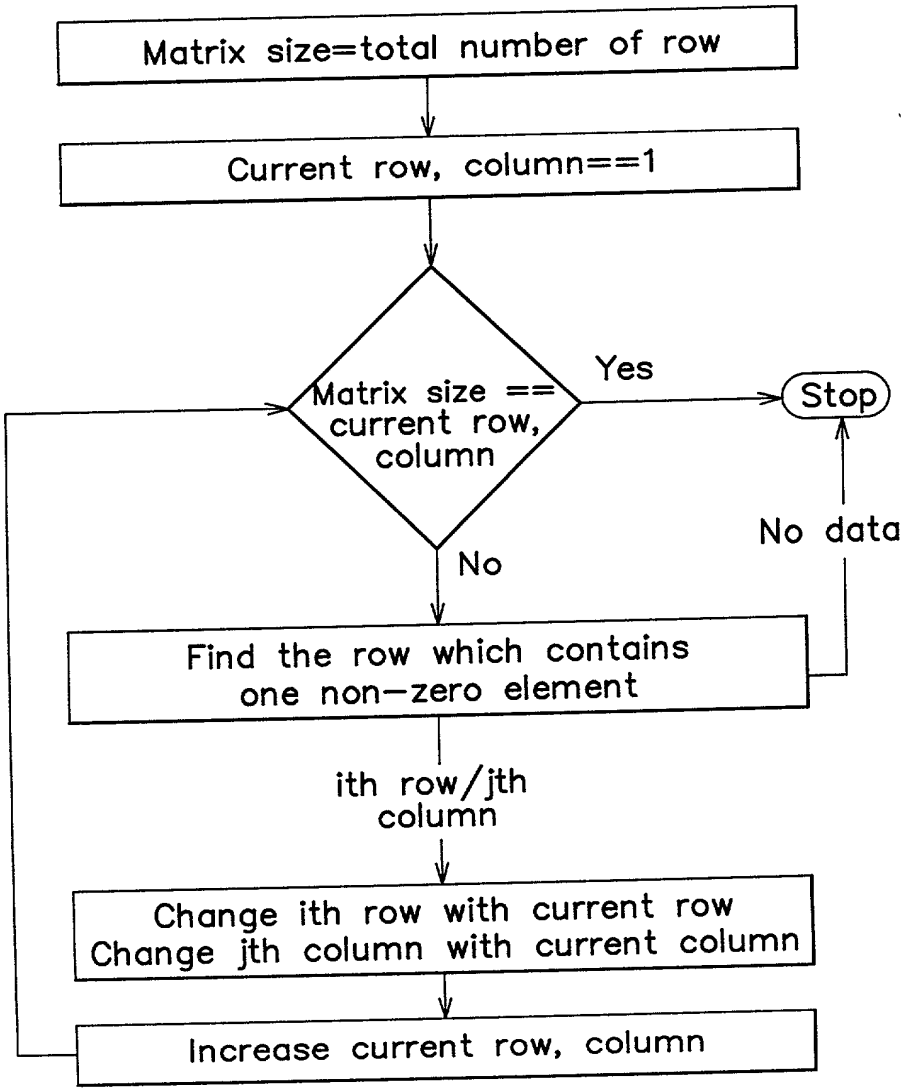

Matrix size=total number of row
Current row, column==1
Matrix size == current row, column
Yes
Stop
No
No data
Find the row which contains one non-zero element
ith row/jth column
Change ith row with current row
Change jth column with current column
Increase current row, column

FIG. 81

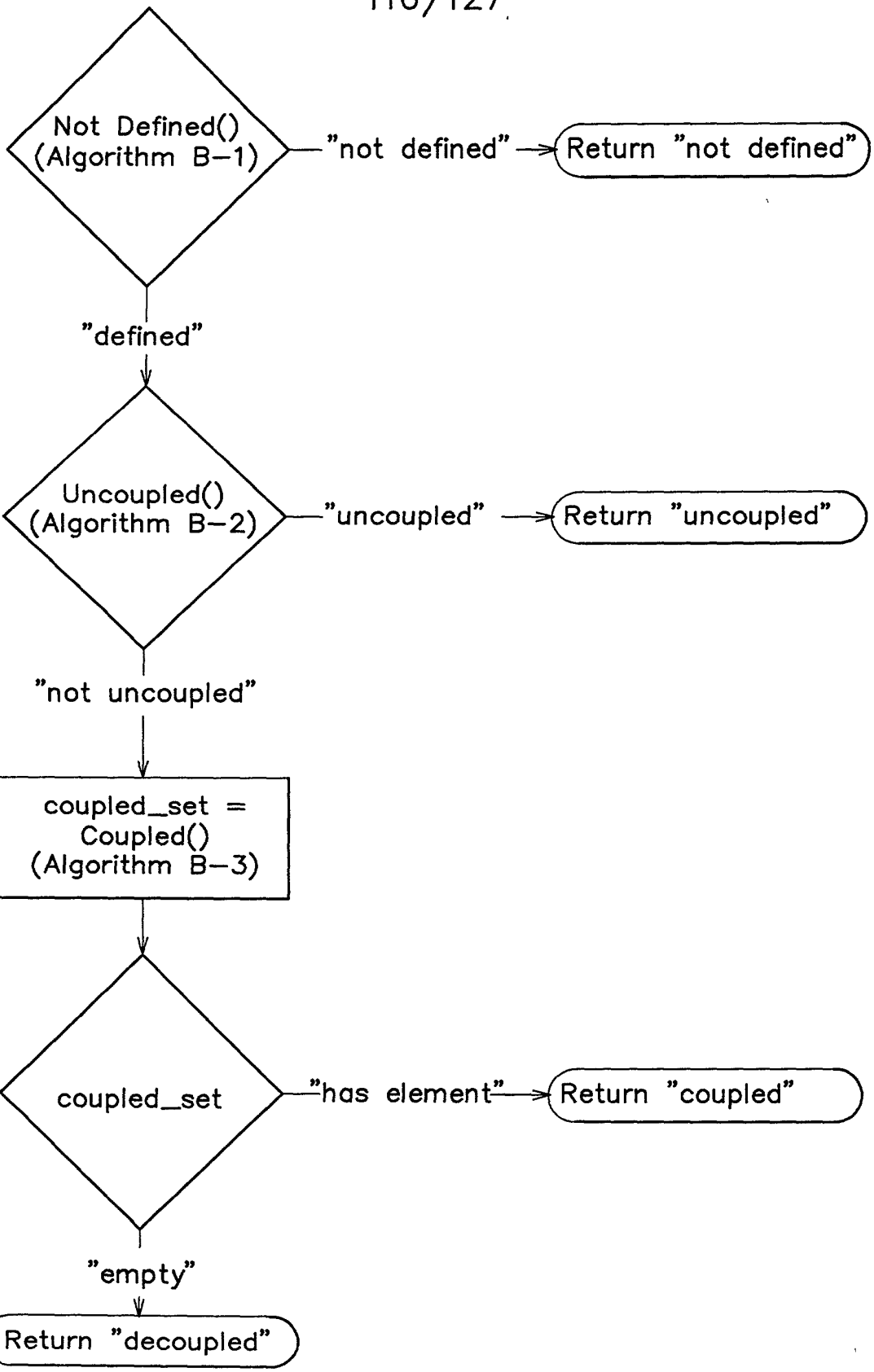

Not Defined()
(Algorithm B–1)
"not defined"
Return "not defined"
"defined"
Uncoupled()
(Algorithm B–2)
"uncoupled"
Return "uncoupled"
"not uncoupled"
coupled_set =
Coupled()
(Algorithm B–3)
coupled_set
"has element"
Return "coupled"
"empty"
Return "decoupled"

FIG. 82

```
Loop One (int row=0; row<total_row_number: row++) {
  Loop Two (int column=0; column <total_column_number; column++) {
    If(maxtrix[row][column] == "empty")
      return "not defined"

    If(row == column) {
      If(matrix[row][column] == "O")
        return "not defined"
    }
  }
}

return "defined"
```

If one of the diagonal element has "0", the design is not defined in terms of the axiomatic design viewpoint

FIG. 83

```
Loop One (int row=0; row<total_row_number: row++) {
  Loop Two (int column=0; column <total_column_number; column++) {
    If(row != column) {
      If(matrix[row][column] == "X")
        return "not uncoupled"
    }
  }
}

return "uncoupled"
```

FIG. 84

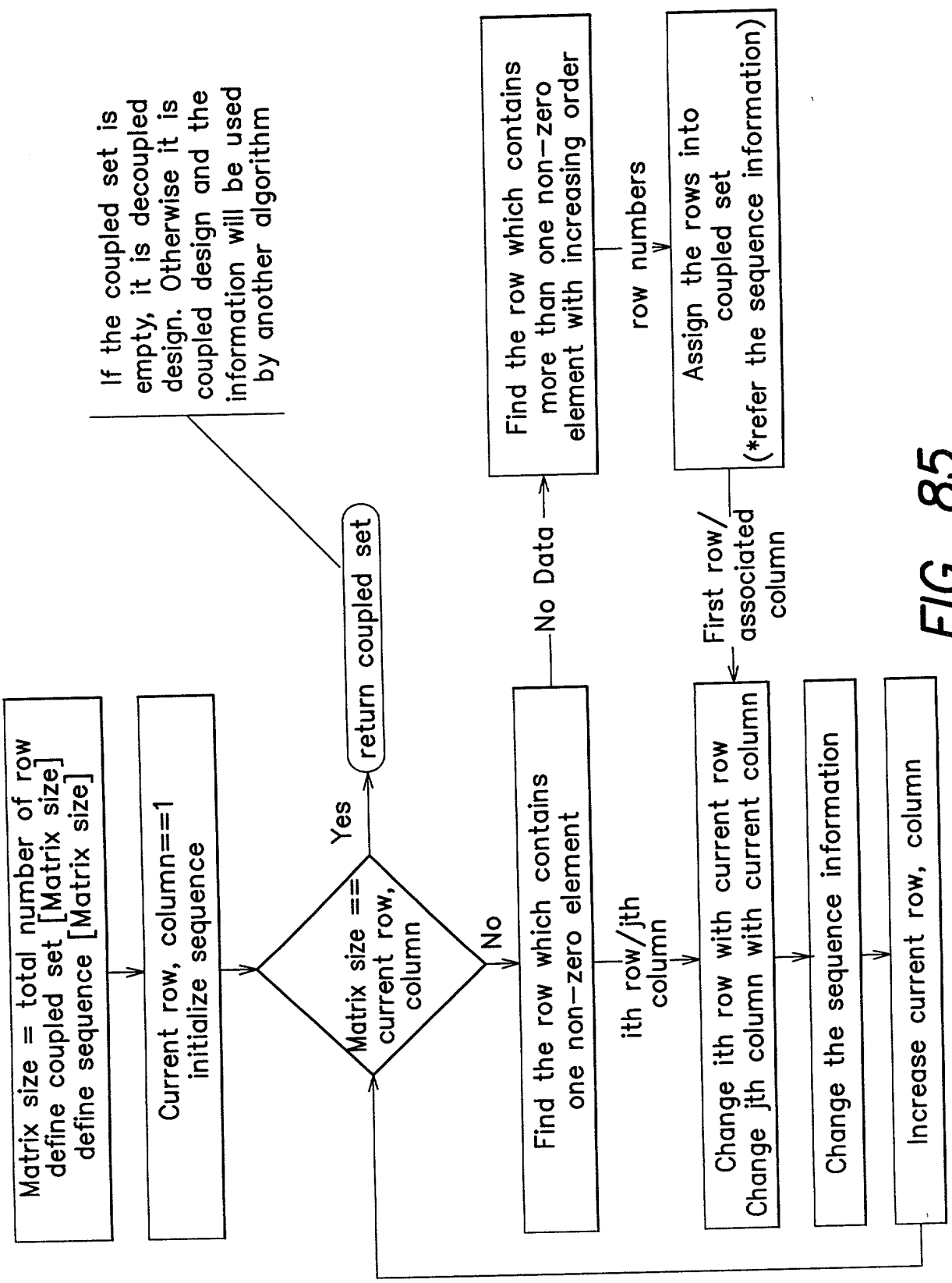

Matrix size = total number of row
define coupled set [Matrix size]
define sequence [Matrix size]
Current row, column==1
initialize sequence
Matrix size == current row, column
Yes
return coupled set
If the coupled set is empty, it is decoupled design. Otherwise it is coupled design and the information will be used by another algorithm
No
Find the row which contains one non-zero element
No Data
Find the row which contains more than one non-zero element with increasing order
row numbers
Assign the rows into coupled set
(*refer the sequence information)
ith row/jth column
First row/ associated column
Change ith row with current row
Change jth column with current column
Change the sequence information
Increase current row, column

FIG. 85

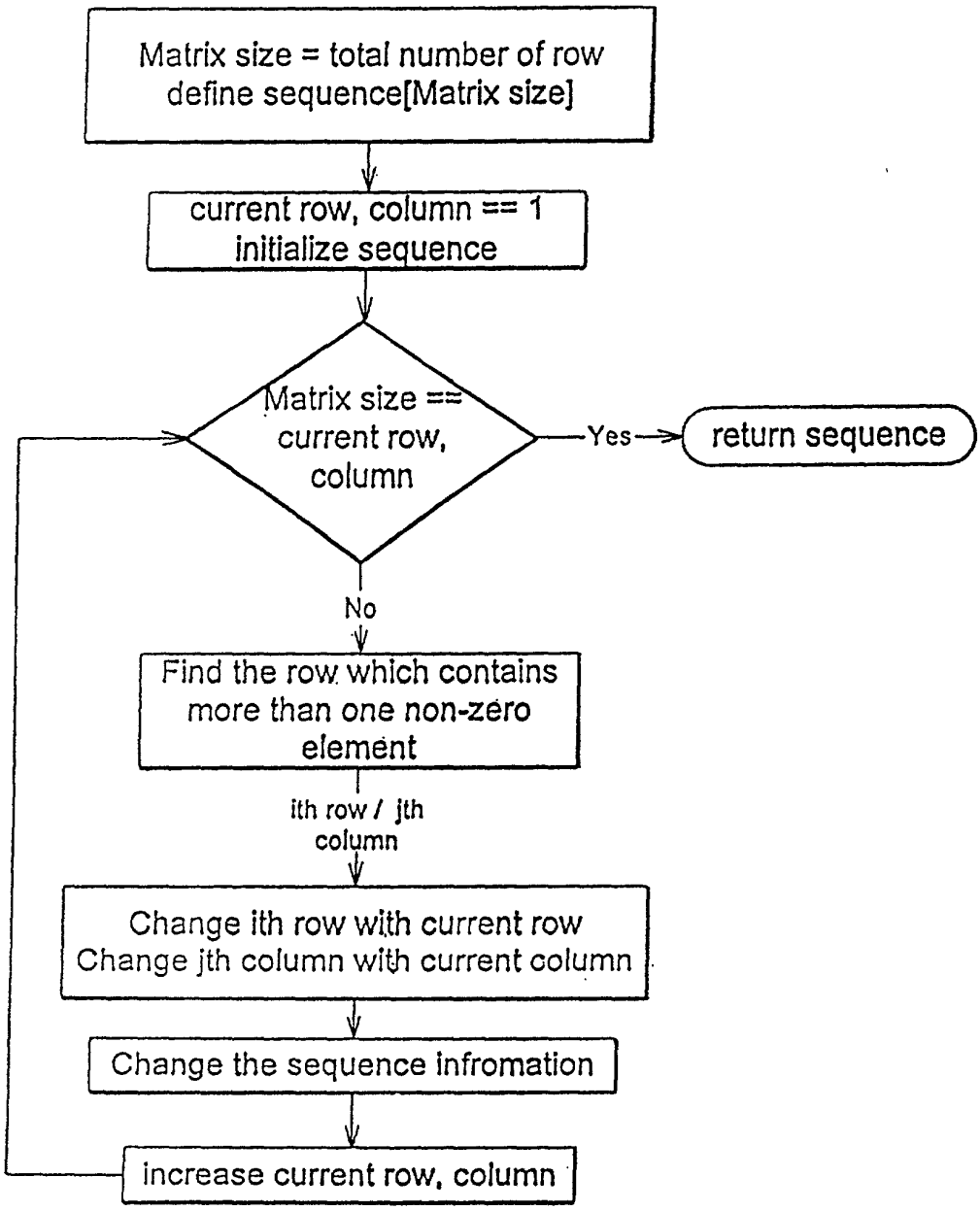

Matrix size = total number of row
define sequence[Matrix size]
current row, column == 1
initialize sequence
Matrix size ==
current row,
column
Yes
return sequence
No
Find the row which contains
more than one non-zero
element
ith row / jth
column
Change ith row with current row
Change jth column with current column
Change the sequence infromation
increase current row, column

FIG. 86

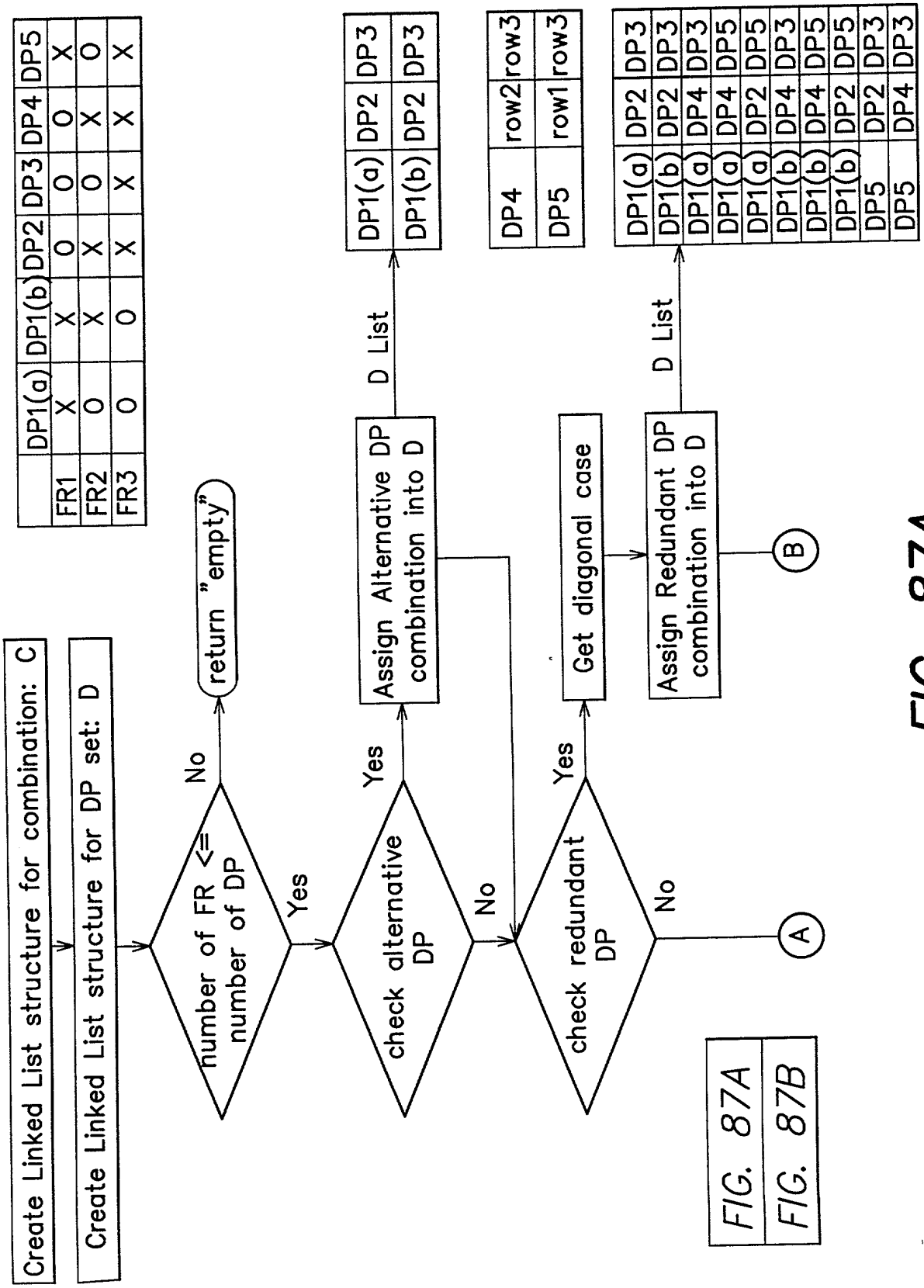


FIG. 87A

FIG. 87A
FIG. 87B

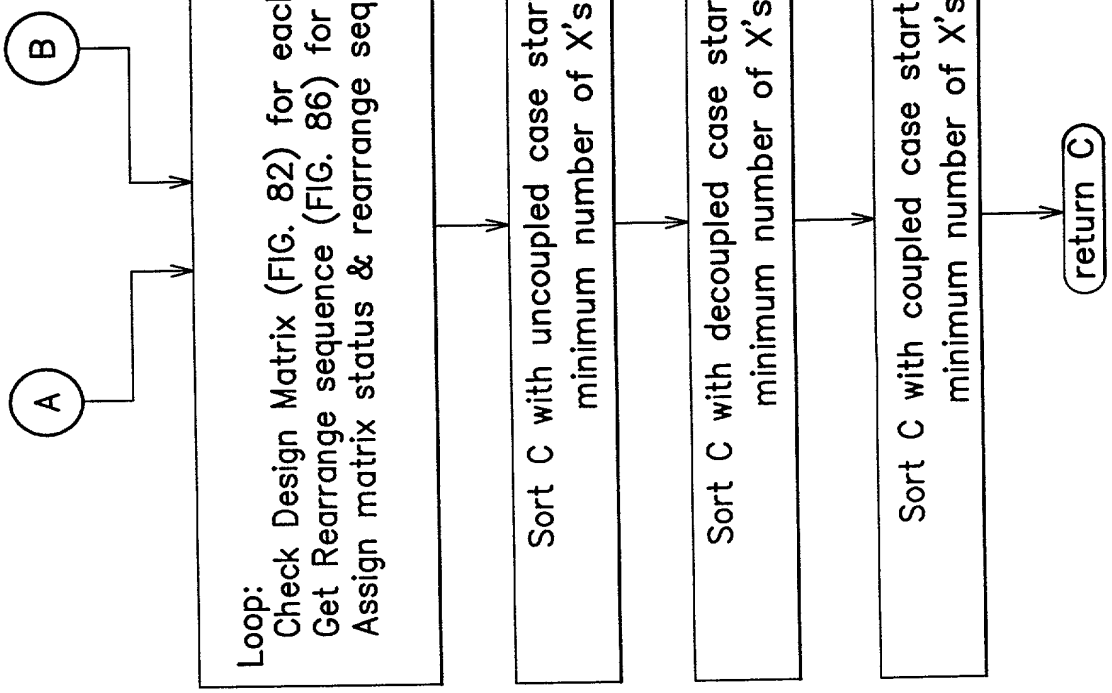
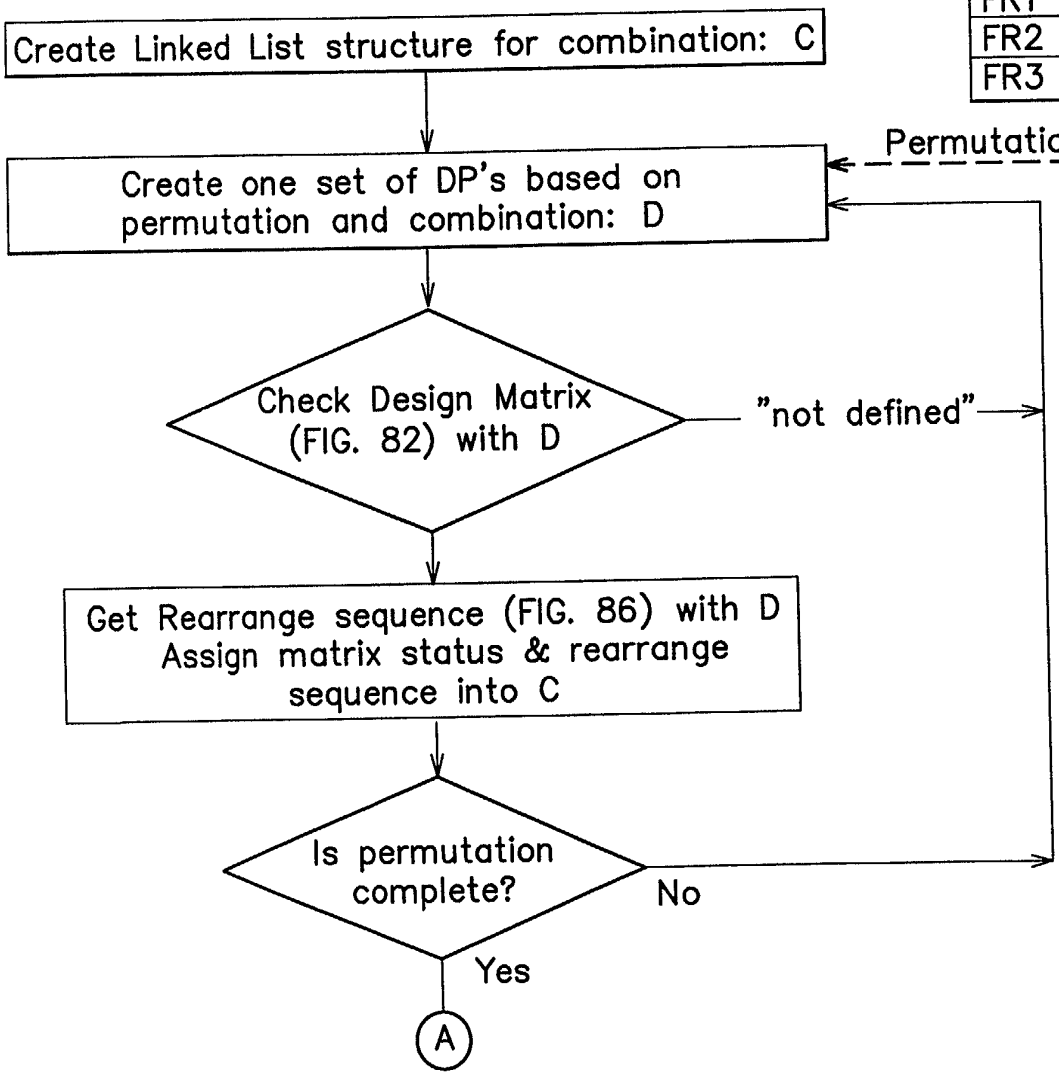

A
B
Loop:
Check Design Matrix (FIG. 82) for each D
Get Rearrange sequence (FIG. 86) for each D
Assign matrix status & rearrange sequence into C
Sort C with uncoupled case starting with minimum number of X's
Sort C with decoupled case starting with minimum number of X's
Sort C with coupled case starting with minimum number of X's
return C

FIG. 87B



	DP1(a)	DP1(b)	DP2	DP3	DP4	DP5
FR1	X	X	O	O	O	X
FR2	O	X	X	O	X	O
FR3	O	O	X	X	X	X

Permutation list

DP1(a)	DP2	DP3
DP1(a)	DP2	DP4
DP1(a)	DP2	DP5
DP1(a)	DP3	DP2
DP1(a)	DP3	DP4
DP1(a)	DP3	DP5
DP1(a	DP4	DP2
...		
...		
DP3	DP4	DP5

FIG. 88A
FIG. 88B

FIG. 88A

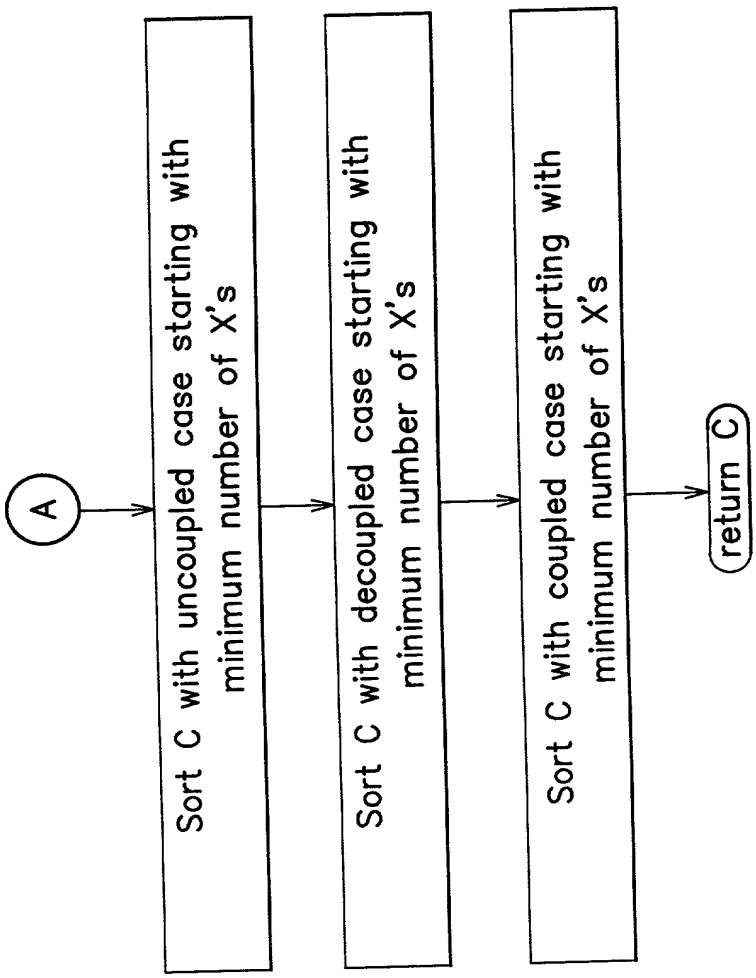

A
Sort C with uncoupled case starting with minimum number of X's
Sort C with decoupled case starting with minimum number of X's
Sort C with coupled case starting with minimum number of X's
return C

FIG. 88B

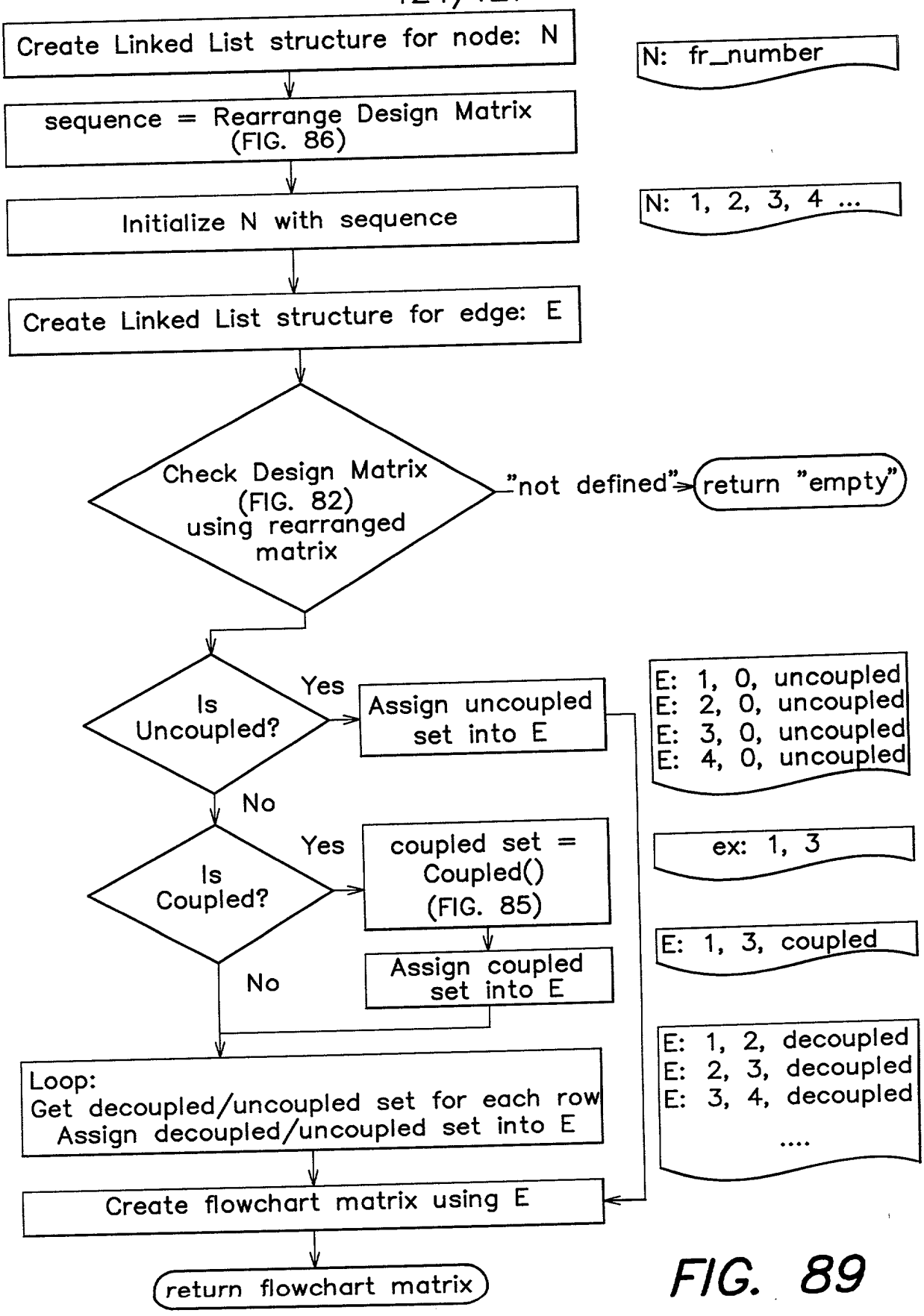

Create Linked List structure for node: N
N: fr_number
sequence = Rearrange Design Matrix (FIG. 86)
Initialize N with sequence
N: 1, 2, 3, 4 ...
Create Linked List structure for edge: E
Check Design Matrix (FIG. 82) using rearranged matrix
"not defined"
return "empty"
Is Uncoupled?
Yes
Assign uncoupled set into E
E: 1, 0, uncoupled
E: 2, 0, uncoupled
E: 3, 0, uncoupled
E: 4, 0, uncoupled
No
Is Coupled?
Yes
coupled set = Coupled() (FIG. 85)
ex: 1, 3
E: 1, 3, coupled
Assign coupled set into E
No
Loop:
Get decoupled/uncoupled set for each row
Assign decoupled/uncoupled set into E
E: 1, 2, decoupled
E: 2, 3, decoupled
E: 3, 4, decoupled
....
Create flowchart matrix using E
return flowchart matrix

FIG. 89

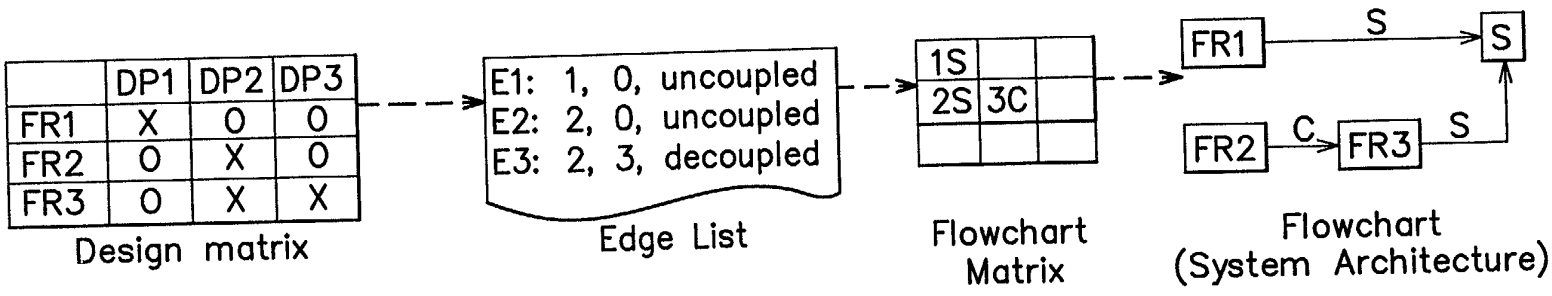


FIG. 90

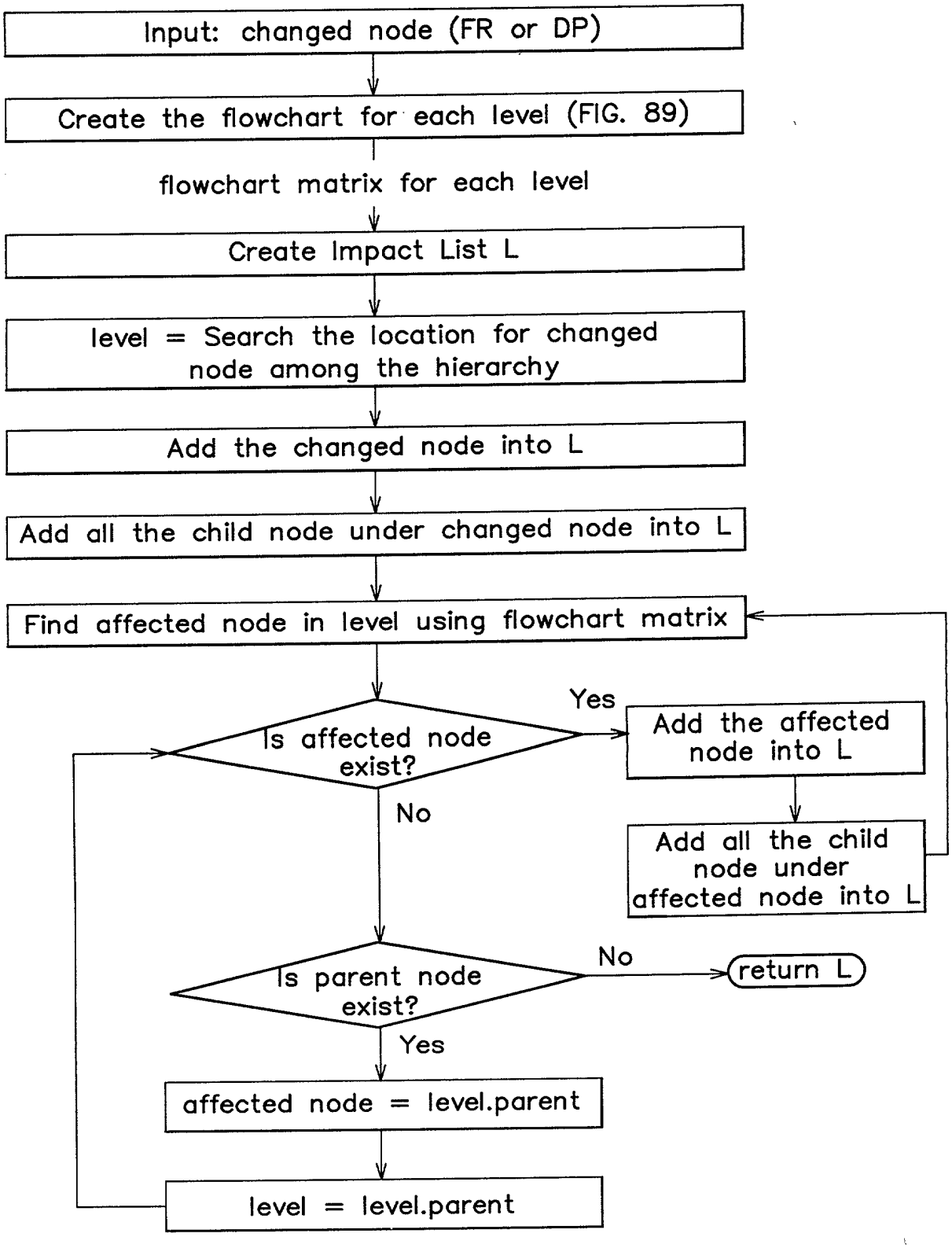

Input: changed node (FR or DP)
Create the flowchart for each level (FIG. 89)
flowchart matrix for each level
Create Impact List L
level = Search the location for changed node among the hierarchy
Add the changed node into L
Add all the child node under changed node into L
Find affected node in level using flowchart matrix
Is affected node exist?
Yes
Add the affected node into L
Add all the child node under affected node into L
No
Is parent node exist?
No
return L
Yes
affected node = level.parent
level = level.parent

FIG. 91

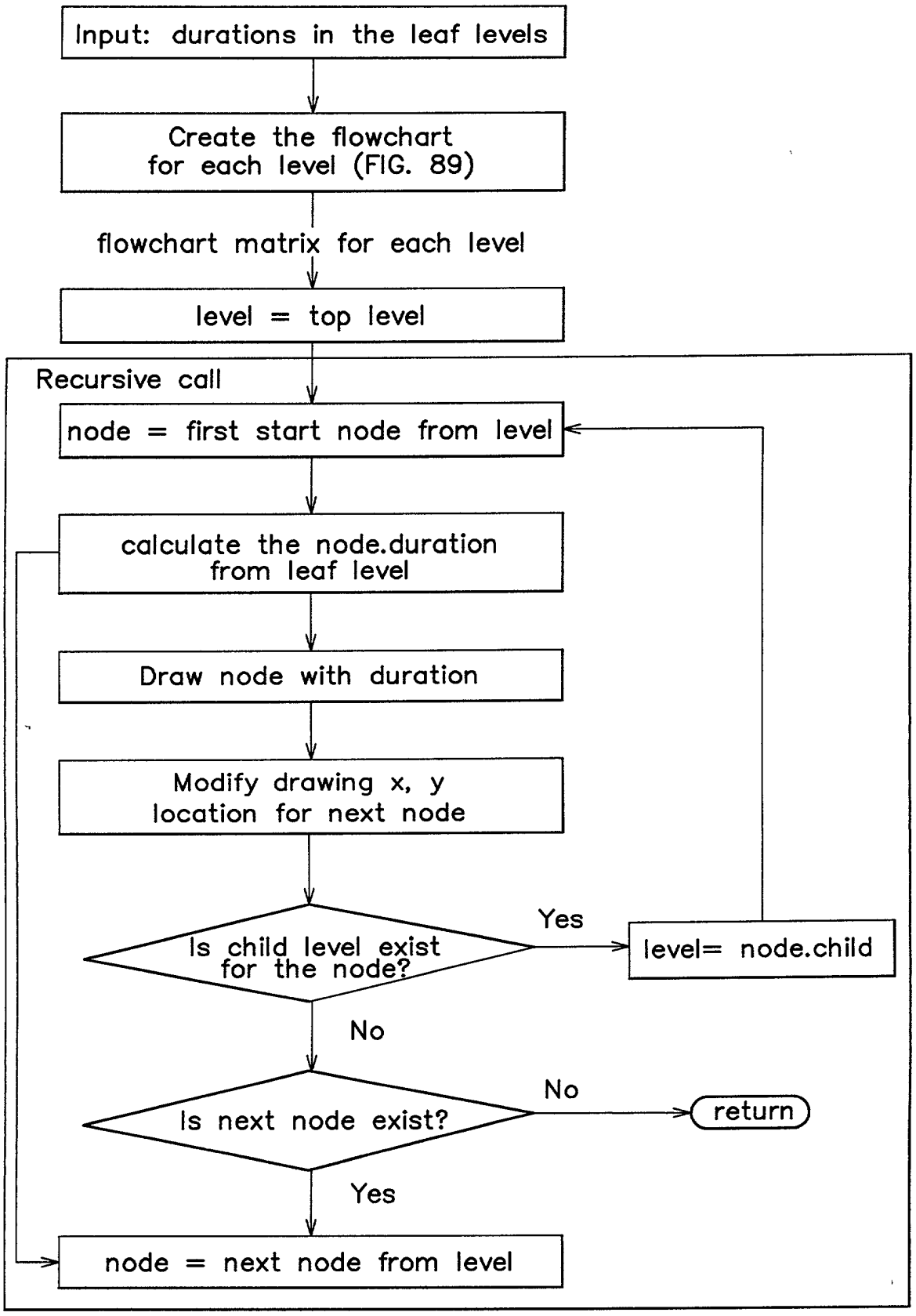

Input: durations in the leaf levels
Create the flowchart for each level (FIG. 89)
flowchart matrix for each level
level = top level
Recursive call
node = first start node from level
calculate the node.duration from leaf level
Draw node with duration
Modify drawing x, y location for next node
Is child level exist for the node?
Yes
level= node.child
No
Is next node exist?
No
return
Yes
node = next node from level

FIG. 92